

UNIVERSIDADE DE LISBOA

Faculdade de Ciências

Departamento de Informática



**INTEGRAÇÃO DE PLATAFORMAS DE APOIO À
APRENDIZAGEM COM OS SISTEMAS DE
INFORMAÇÃO ACADÉMICOS DA UNIVERSIDADE
DE LISBOA**

Francisco Guedes Lebre Bruno Soares

DISSERTAÇÃO

MESTRADO EM ENGENHARIA INFORMÁTICA
Especialização em Arquitectura, Sistemas e Redes de Computadores

2014

UNIVERSIDADE DE LISBOA

Faculdade de Ciências

Departamento de Informática



**INTEGRAÇÃO DE PLATAFORMAS DE APOIO À
APRENDIZAGEM COM OS SISTEMAS DE
INFORMAÇÃO ACADÉMICOS DA UNIVERSIDADE
DE LISBOA**

Francisco Guedes Lebre Bruno Soares

DISSERTAÇÃO

MESTRADO EM ENGENHARIA INFORMÁTICA
Especialização em Arquitectura, Sistemas e Redes de Computadores

Dissertação orientada pelo Prof. Doutor Hugo Alexandre Tavares Miranda

2014

Agradecimentos

Quero deixar um agradecimento a todos aqueles que acompanharam e apoiaram o meu percurso académico. Um agradecimento especial à minha família que me apoiou imenso na concretização dos objectivos que tenho vindo a traçar ao longo do tempo e que sempre me motivou naquilo que realmente gosto de fazer. Obrigado Mãe e Pai pelo esforço para que tudo fosse possível.

Quero deixar um agradecimento especial ao meu orientador Hugo Miranda pelo apoio na concretização da minha dissertação e pelo exemplo de esforço e dedicação durante o período em que trabalhámos.

Aos meus colegas de Faculdade, nomeadamente Gonçalo Esteves, João Cruz, Tiago Pereira, João Carregosa, Rui Gonçalves, Henrique Vaz e João Feio cujo esforço, dedicação e motivação foram um exemplo a seguir ao longo deste percurso.

Aos meus amigos que com muita dedicação e paciência estiveram sempre presentes. Obrigado João Vieira, Vitor Martins, Luís Miguel Reis, Patrícia Tavares e Gonçalo Cardoso.

Por último, à minha actual *dream team* com quem tenho partilhado grandes momentos de criação e desenvolvimento pessoal Tiago Tresca, Duarte Costa, Pedro Simão, Andreia Jacques e Miguel Maldonado um muito obrigado pelo vosso apoio nos últimos meses.

Dedicatória.

Resumo

A utilização de diferentes plataformas para a gestão de informação e publicação de conteúdos nas Escolas da Universidade de Lisboa tem vindo a levantar problemas de coerência dos dados e duplicação de informação e esforço.

Este trabalho aborda os desafios colocados pela integração da informação entre duas plataformas com objetivos e modelos de desenvolvimento distintos, uma de gestão académica SiGES e a outra de apoio à aprendizagem Moodle.

A comunicação entre as plataformas foi assegurada pela tecnologia de serviços web, beneficiando desta forma da existência de mecanismos básicos de suporte a esta tecnologia utilizada em ambas.

Além de diagnosticar novos problemas e resolver alguns dos já identificados, este trabalho define um modelo que facilitará a concretização de novas funcionalidades e automatismos para outras necessidades que se venham a colocar no futuro.

Palavras-chave: Integração de Sistemas de Informação, Serviços Web, Moodle, SiGES

Abstract

With the use of different platforms for information management and publishing content in Schools of University of Lisbon there have been raised some issues of data consistency and duplication of information and effort.

This paper addresses the challenges posed by the integration of information between two platforms with different objectives and models of development including the academic management platform SiGES and the learning management platform Moodle.

The communication between the two platforms was provided by web services technology, thus benefiting from the existence of basic mechanisms supporting this technology on both platforms.

In addition to identify and resolve some of the problems already identified, this work defines a model that will facilitate the implementation of new features and automatic procedures for other needs that may be challenged in the future.

Keywords: Integration of Information Systems, Web Services, Moodle, SiGES

Conteúdo

Lista de Figuras	xvi
Lista de Tabelas	xix
1 Introdução	1
1.1 Motivação	1
1.2 Objectivos	1
1.3 Estrutura do documento	2
1.4 Contribuições	2
2 Plataformas e Tecnologias	5
2.1 Plataformas	5
2.1.1 Moodle	5
2.1.2 SiGES	5
2.2 Tecnologias utilizadas	7
2.2.1 Serviços Web	7
2.2.2 Sistemas em Produção	8
3 Análise	11
3.1 Entidades envolvidas	11
3.2 Plataformas	12
3.2.1 Moodle	12
3.2.2 SiGES	13
3.2.3 Requisitos funcionais	13
4 Desenho	19
4.1 Histórico Moodle	19
4.2 Apresentação e Organização das Unidades Curriculares	20
4.3 MOODLE-IS com Projectão Multi-instâncias	22
4.3.1 Ponto de situação	23
4.3.2 Solução proposta	24
4.4 Comunicação entre plataformas	25

4.4.1	Modelo de Comunicação	25
4.4.2	Comunicação de sumários entre plataformas	26
4.4.3	Comunicação de classificações entre plataformas	27
4.4.4	Dados a guardar no serviço web	28
4.4.5	Desenho do cliente do serviço web	28
4.5	Interfaces dos módulos Moodle	29
4.5.1	Módulo Degrees	29
4.5.2	Módulo SummaryWS	30
4.5.3	Módulo GraderWS	31
5	Desenvolvimento	39
5.1	Histórico Moodle	39
5.2	Módulo Degrees	42
5.2.1	Concretização da Base de Dados	42
5.2.2	Adaptação da interface	42
5.3	MOODLE-IS	42
5.3.1	Alterações dif-elearning	43
5.3.2	Alterações MOODLE-IS	43
5.4	Concretização da comunicação entre plataformas	43
5.4.1	Base de Dados de testes	43
5.4.2	Concretização do Serviço Web	44
5.4.3	Módulo SummaryWS	45
5.4.4	Módulo GraderWS	45
6	Avaliação	55
6.1	Histórico Moodle	55
6.2	Apresentação e Organização das Unidades Curriculares	55
6.3	MOODLE-IS com Projecção Multi-Instâncias	56
6.4	Comunicação entre plataformas	56
7	Conclusão	59
7.1	Histórico Moodle	59
7.2	Comunicação entre Moodle e SiGES	59
	Bibliografia	64

Lista de Figuras

3.1	Mapa de Fluxo de Trabalho sobre Sumários	15
3.2	Mapa de Fluxo de Trabalho para solução proposta sobre Sumários	16
3.3	Mapa de Fluxo de Trabalho sobre Classificações	16
3.4	Mapa de Fluxo de Trabalho para solução proposta sobre Classificações . .	17
4.1	Diagrama de Operações para Transferência de Disciplinas entre instâncias Moodle	21
4.2	Modelo de Dados de Categorias no Moodle	22
4.3	Modelo de Dados do Módulo para Categorias	22
4.4	Diagrama de Operações para Projectar Disciplinas SiGES no MOODLE .	23
4.5	Diagrama de Operações para Projectar Disciplinas SiGES em várias plataformas Moodle	24
4.6	Parte do Diagrama de Classes dif-elearning	25
4.7	Alterações ao Diagrama de Classes dif-elearning	26
4.8	Fluxo de operações para apresentação de Sumários	27
4.9	Fluxo de operações para escrita de Sumários	28
4.10	Comunicação de sumários com Serviço Web	29
4.11	Mensagens necessárias para comunicação de Sumários com Serviço Web	30
4.12	Fluxo de operações para exportação de Classificações	31
4.13	Comunicação de Classificações com Serviço Web	32
4.14	Mensagens necessárias para comunicação de Classificações com Serviço Web	33
4.15	Modelo de Dados do Serviço Web	34
4.16	Comunicação de Sumários	34
4.17	Criação de uma Categoria	35
4.18	Projecção de uma Unidade Curricular numa Categoria	35
4.19	Lista de Degrees	35
4.20	Unidades Curriculares de um Degree	35
4.21	Lista de Sumários	35
4.22	Escrita de um Sumário	36
4.23	Interface para Configuração de Projecções das Componentes de Avaliação	36
4.24	Mensagem de Sucesso da avaliação das Projecções de Classificações . . .	37

4.25	Mensagem de Erro da avaliação das Projeções de Classificações	37
4.26	Resultado do Processo de Exportação de Classificações	37
5.1	Script de Contexto para o Backup de Unidades Curriculares	40
5.2	Introdução do Script de Backup	40
5.3	Script para Execução do Processo	41
5.4	Introdução do Script de Backup	41
5.5	ServiceDAO.java	46
5.6	Definições de tipos XSD para Classificações	47
5.7	Definições de tipos XSD para Sumários	47
5.8	Definições de tipos XSD para Mensagens das Classificações	48
5.9	Definições de tipos XSD para Mensagens das Classificações	48
5.10	Definições de tipos XSD para Mensagens dos Sumários	49
5.11	Definições de tipos XSD para Mensagens dos Sumários	49
5.12	Definições de WSDL Port Types para Classificações	50
5.13	Definições de WSDL Messages para Classificações	50
5.14	Definições de WSDL Bindings para Classificações	51
5.15	Definições de WSDL Port Types para Sumários	52
5.16	Definições de WSDL Messages para Sumários	52
5.17	Definições de WSDL Bindings para Sumários	53

Lista de Tabelas

2.1	Definição WSDL	8
5.1	Utilização exemplo rsync	42
5.2	Configuração frontpage Moodle	42

Capítulo 1

Introdução

1.1 Motivação

Na Universidade de Lisboa a gestão de informação académica e gestão de conteúdos são assegurados por dois sistemas independentes.

O sistema de gestão de informação académica da instituição (MIS, do inglês Management Information System) de nome SiGES, foi produzido e é mantido pela Digitalis. Tem como propósito funcional a gestão de informação referente à organização.

O sistema de apoio à aprendizagem que tem por objectivo fornecer mecanismos de partilha de recursos e gestão de actividades com os alunos (LMS, do inglês Learning Management System) denomina-se Moodle.

Como os dois sistemas foram anteriormente implementados em separado, não foi possível a sua integração numa única plataforma. Esta separação é inconveniente uma vez que obriga em alguns casos à duplicação de informação e noutros à necessidade de proceder à navegação entre múltiplos sistemas para aceder ou processar a informação relacionada.

1.2 Objectivos

Este trabalho, enquadrado num programa de estágio na equipa de desenvolvimento dos Serviços Centrais da Universidade de Lisboa, apresenta uma contribuição para a integração de informação entre estes dois sistemas. Teve como objectivo reduzir os fluxos de trabalho redundantes dos utilizadores e administradores, por forma a oferecer uma maior motivação para a sua utilização. Isto é, pretende-se fortalecer a ligação entre a plataforma de apoio à aprendizagem e o sistema de informação académico de modo a que alguma da informação relevante para ambas as plataformas seja submetida apenas uma vez e possa ser consultada em qualquer um dos sistemas.

Durante a conceptualização e desenvolvimento deste trabalho existiu uma preocupação na sua documentação por forma a ser reutilizado. É fundamental que as técnicas utilizadas

possam ser aprofundadas e do conhecimento de qualquer pessoa que exerça responsabilidade na manutenção e desenvolvimento dos sistemas em questão.

É, também, um dos objectivos principais que, mesmo com processos automáticos de interacção, estes sistemas ofereçam grande disponibilidade, coerência e fiabilidade da informação e, por isso, serão requisitos a ter em conta durante a concretização do trabalho.

1.3 Estrutura do documento

Este documento está dividido em sete capítulos organizados da seguinte forma:

Capítulo 2: Plataformas e Tecnologias Utilizadas No segundo capítulo deste trabalho são apresentadas as plataformas onde o problema é levantado, quais as tecnologias em que estas foram desenvolvidas e quais as tecnologias que foram tomadas em conta para a concretização dos objectivos apresentados.

Capítulo 3: Análise No capítulo de análise foram feitos levantamentos sobre as entidades envolvidas para a possível concretização do trabalho e como funcionam actualmente as plataformas face aos problemas identificados seguindo-se um levantamento dos requisitos junto das entidades envolvidas.

Capítulo 4: Desenho Neste capítulo são apresentadas as soluções consideradas para os requisitos levantados no capítulo anterior. Esta fase recorre ao desenho técnico por forma a compreender como estão actualmente implementadas as plataformas, onde podem ser concretizadas as soluções e o que será necessário para o desenvolvimento das mesmas.

Capítulo 5: Desenvolvimento No capítulo de desenvolvimento são apresentadas considerações técnicas para a concretização das soluções tendo em conta os requisitos levantados.

Capítulo 6: Avaliação Neste capítulo são apresentadas as técnicas de avaliação do trabalho e os resultados das soluções propostas, isto é, da forma como foram as soluções concretizadas e os resultados conseguidos.

Capítulo 7: Conclusões Por último, são apresentadas as considerações finais com propostas da melhoria de algumas soluções e para a continuação do trabalho.

1.4 Contribuições

Este trabalho apresenta as seguintes contribuições para a simplificação das actividades realizadas por alunos e docentes no âmbito da gestão académica e pedagógica na Univer-

sidade de Lisboa:

1. Integração dos sistemas de gestão de classificações dos alunos
2. Integração dos sistemas de gestão dos sumários das aulas leccionadas pelos Docentes
3. Organização da apresentação das Unidades Curriculares aos alunos
4. Gestão de conteúdos das Unidades Curriculares de anos lectivos anteriores

Capítulo 2

Plataformas e Tecnologias

Na realização deste trabalho foi utilizado um conjunto de plataformas e tecnologias com vista à concretização dos objectivos apresentados.

2.1 Plataformas

2.1.1 Moodle

O Moodle¹ é uma plataforma de apoio à aprendizagem que se encontra actualmente em produção na Universidade de Lisboa. O objectivo desta plataforma é facilitar a interacção dos docentes com os alunos, utilizando metodologias pedagógicas para auxiliar o ensino e aprendizagem virtual ou semi-presencial². No âmbito do desenvolvimento do Moodle foi feita uma análise da capacidade do mesmo para a integração de Módulos. O Moodle é desenvolvido em PHP³ e dispõe de uma API[8] (do inglês, Application Programming Interface) organizada para o desenvolvimento de módulos[10] que permitem adicionar funcionalidades. Estes módulos são organizados de acordo com o tipo de funcionalidades que se querem desenvolver e disponibilizam metodologias para manipulação das várias componentes que a plataforma integra: configurações, interface, gestão de acesso, modelo de dados e até funcionalidades próprias da plataforma, como por exemplo a gestão de conteúdos e avaliações.

2.1.2 SiGES

A plataforma SiGES⁴ faz a gestão de informação referente a funcionários, docentes e alunos numa instituição e que envolve actividades como a distribuição de serviço docente, controlo do sistema de horários, lançamento de notas, gestão de sumários, seriação de

¹<http://moodle.org>

²http://pt.wikipedia.org/wiki/Learning_management_system

³<http://php.net>

⁴http://www.digitallis.pt/index.php?option=com_content&view=article&id=59&Itemid=56

candidatos, entre outros. No contexto do SiGES estas funcionalidades são divididas em módulos, instalados consoante os requisitos de cada instituição.

O SiGES foi desenvolvido numa framework de código aberto J2EE, criada pela Digitalis⁵, de nome DIF. A DIF é uma plataforma de trabalho para a criação de aplicações web pensada com os principais objectivos de promover um ambiente de desenvolvimento *component-driven* e uma arquitectura modular que potencia integrações e configurações tanto a nível de lógica de negócio como de interface de utilização. Esta estratégia favorece a reutilização e a adaptação, parcial ou totalmente, de soluções já concretizadas para resolver novos problemas.

A versão 2.0 da DIF[1], de nome DIF2 e que é actualmente utilizada, foi desenvolvida segundo um padrão de programação de nome *Inversion of Control* (IoC)⁶. Em contraste com o padrão tradicional de programação, onde existe o propósito de invocar bibliotecas reutilizáveis de código para execução de tarefas genéricas do programa, na IoC é o código reutilizável que invoca o código configurável, isto é, específico ao negócio.

A DIF2 é dividida em quatro componentes principais:

DIF Core onde é controlada toda a lógica do funcionamento da plataforma[2];

DIF Model é responsável pela gestão do modelo de dados da plataforma com recurso a tecnologias de base de dados como o *Hibernate*[3];

DIF Presentation é responsável por toda a camada de apresentação web, suportando diversos formatos (HTML, JSON ou XML) com recurso a JSP Pages e Javascript[5];

DIF Modules são um conjunto de módulos adicionais que acrescentam funcionalidades à plataforma, como por exemplo a implementação de protocolos de Autenticação ou a integração da própria plataforma numa versão anterior[4].

Graças à sua arquitectura modular a DIF2 dispõe também de projectos satélites, isto é projectos que fornecem ferramentas úteis mas cujo funcionamento é independente da plataforma. O *digi-utils*[6], por exemplo, é um desses projectos que foi utilizado na construção da própria plataforma.

Para a compilação da Framework e dos seus módulos a Digitalis utiliza uma ferramenta de nome *Maven*⁷. Esta ferramenta oferece agilidade na gestão de dependências, compilação, distribuição e desenvolvimento em colaboração por via de repositórios partilhados. É definido por um conjunto de regras e um software usado para gerir e descrever projectos que seguem as regras *Maven*. Define um ciclo para compilação, testes e publicação de artefactos, também chamados de módulos. *Maven* é um projecto

⁵<http://www.digitalis.pt/>

⁶http://en.wikipedia.org/wiki/Inversion_of_control

⁷<http://maven.apache.org/>

proprietário da *Apache Software Foundation* e é distribuído com uma licença de código aberto.

Para o desenvolvimento foram utilizadas as ferramentas disponibilizadas pelo *Eclipse*⁸ na sua versão *Kepler Java Development Environment* configurada de acordo com um conjunto de regras da Digitalis para organização e documentação do código.

2.2 Tecnologias utilizadas

2.2.1 Serviços Web

Os Serviços Web são uma solução popular na comunicação entre aplicações web. Através duma interface bem definida, os serviços web permitem a transferência de dados sobre HTTP. O objectivo desta tecnologia é que uma aplicação possa invocar outra para efectuar tarefas independentemente da plataforma ou da linguagem de programação de cada uma. A tecnologia descreve os recursos utilizando XML (do inglês eXtensible Markup Language)⁹ ou JSON (do inglês JavaScript Object Notation)¹⁰ e identifica-os por um URI (do inglês Uniform Resource Identifier)¹¹. Foram consideradas duas tecnologias possíveis para a concretização de serviços web neste trabalho.

SOAP, do inglês Simple Object Access Protocol que define mais que uma arquitectura, é um protocolo baseado em XML para o formato das mensagens de um serviço e que assenta em protocolos da camada de aplicação RPC e HTTP para a comunicação.

Ao contrário de SOAP, REST, do inglês Representative State Transfer, é uma arquitectura que não define o formato das suas mensagens e que assenta na comunicação por HTTP para as operações de CRUD (do inglês Create, Remove, Update Data). O REST não é um protocolo específico de comunicação sendo que as mensagens podem utilizar XML, JSON ou até texto simples.

A grande diferença entre estas duas arquitecturas é o grau de acoplamento entre o cliente e o servidor. SOAP é um protocolo com regras bem definidas e com um contracto de comunicação que torna o cliente bastante acoplado ao servidor, ao contrário da arquitectura REST que por não assentar em qualquer contracto torna o cliente mais independente do servidor. *Um cliente REST é como um browser. É um cliente genérico que sabe como utilizar um protocolo e os seus métodos e a aplicação tem que se enquadrar nessa lógica*¹². Ao contrário de um cliente SOAP, um cliente REST pode comunicar com uma API (do inglês Application Programming Interface) sem conhecimento nenhum à partida, enquanto no SOAP o cliente deve conhecer todo o seu contracto, o que pode e o que não pode fazer, antes de qualquer chamada ao serviço.

⁸<https://www.eclipse.org/>

⁹<http://www.w3.org/XML/>

¹⁰<http://json.org/>

¹¹<http://www.w3.org/Addressing/>

¹²<http://stackoverflow.com/questions/19884295/soap-vs-rest-differences>

WSDL Ports	Descreve um serviço web, as operações que podem ser realizadas e as mensagens envolvidas.
WSDL Messages	Define a organização das mensagens a comunicar.
WSDL Types	Definem os tipos de dados usados nas mensagens
WSDL Bindings	Define as mensagens utilizadas em cada operação.

Tabela 2.1: Definição WSDL

WSDL, do inglês Web Services Description Language¹³, é uma linguagem baseada em XML que descreve serviços web como um conjunto de pontos de entrada sobre operações que contém informação representativa documental ou procedimental. É a linguagem que permite definir um contracto de comunicação no protocolo SOAP. Utiliza um conjunto de elementos bem definido, representado na Tabela 2.1.

No que diz respeito à disponibilização de serviços web pelas plataformas em questão, o Moodle na versão 2[13] e o SiGES¹⁴ dispõem ambos dos seus módulos para a integração de serviços web sobre a interface de aplicação dos seus sistemas.

Para a concretização dos serviços web utilizou-se um ambiente de desenvolvimento com base na linguagem JAVA com recurso à tecnologia de serviços web JAX-WS¹⁵, a executar num servidor JBoss¹⁶ versão 7.1. Para a instalação de um ambiente de testes foram utilizadas as ferramentas disponibilizadas pelo Eclipse¹⁷ na sua versão *Kepler Java Development Environment*.

Por fim, para o desenho da base de dados necessária para o serviço de testes foi utilizada a ferramenta *MySQL Workbench*¹⁸ que permite, através de desenho, a produção eficiente de código MySQL.

2.2.2 Sistemas em Produção

No âmbito do projecto "e-learning na UL" existe uma plataforma Moodle na versão 1.9 disponível em <http://elearning.ul.pt> que aloja unidades curriculares de 10 instituições de ensino da Universidade de Lisboa. Esta plataforma é suportada por um servidor com 8 cores, Intel(R) Xeon(R) CPU E5345 @ 2.33GHz, 16243 Mb de RAM, a correr Linux

¹³<http://www.w3.org/TR/wsdl>

¹⁴http://www.digitalis.pt/index.php?option=com_content&view=category&id=103%3Awebservices-digitalis&Itemid=75

¹⁵<https://jax-ws.java.net/>

¹⁶<http://www.jboss.org/>

¹⁷<https://www.eclipse.org/>

¹⁸<http://www.mysql.com/products/workbench/>

CentOS release 5.10 (Final) com Apache/2.2.3, MySQL Versão 14.14 Distribuição 5.5.30. Este servidor é identificado pelo nome ul-moodle01.ul.pt.

Para a Faculdade de Farmácia foi instalada uma plataforma Moodle 2.4 num servidor Intel(R) Xeon(R) DualCore CPU X5560 @ 2.80GHz, 1877 Mb de RAM, a correr um Linux CentOS release 6.4 (Final) com Apache/2.2.15, MySQL Versão 14.14 Distribuição 5.1.69. O servidor é identificado pelo nome ul-moodle03.ul.pt.

Ambas as plataformas estão ligadas a um agregador de autenticação da Universidade de Lisboa. Como clientes deste serviço, a ul-moodle01 tem configurado um Simple-SamlPHP¹⁹ e a ul-moodle03 um Shibboleth²⁰ na versão 2.5.2.

¹⁹<http://simplesamlphp.org/>

²⁰<http://shibboleth.net/>

Capítulo 3

Análise

3.1 Entidades envolvidas

As responsabilidades de desenvolvimento, implementação, manutenção e formação na utilização dos sistemas estão divididas por diversas unidades da Universidade de Lisboa.

O *e-learning lab* é um projecto que surge em Julho de 2010, agregado ao Instituto da Educação, para promover a integração de tecnologias nas práticas docentes. Tem como objectivos divulgar e generalizar a adopção da plataforma de apoio à aprendizagem da Universidade de Lisboa, sensibilizar e capacitar os docentes para a utilização da mesma e fomentar e fornecer apoio na adaptação e migração de conteúdos curriculares.¹ Neste projecto o *e-learning lab* contribui para o levantamento e definição de requisitos funcionais e não funcionais na óptica do utilizador da plataforma de apoio à aprendizagem.

A *Unidade de Serviços Centrais da Universidade de Lisboa* engloba os Serviços Tecnológicos responsáveis pelas áreas de sistemas e comunicações, sistemas de informação e de apoio informático e disponibiliza o alojamento e manutenção dos sistemas em questão.²

A *Administração de Sistemas do Departamento de Informática da Faculdade de Ciências*³ (ADMIN DI) é a unidade responsável pela manutenção e desenvolvimento da rede e dos sistemas desse departamento. Actualmente o *DI FCUL* dispõe de uma plataforma de apoio à aprendizagem Moodle, na qual estão integradas algumas funcionalidades desenvolvidas localmente e contribui para a análise, desenvolvimento e implementação deste trabalho.

A *Faculdade de Farmácia* é a instituição académica onde, por sugestão do *e-learning lab*, será implementada uma primeira concretização deste trabalho.

A *Digitalis* é uma empresa de software e serviços que surgiu em 1994 com o principal objectivo de servir as instituições de ensino nas mais variadas tecnologias de gestão através de instalação e manutenção de sistemas informáticos abrangentes. Estudantes, administradores, docentes, colaboradores, funcionários, antigos alunos e potenciais can-

¹<http://elearninglab.ul.pt>

²<http://www.sp.ul.pt>

³<http://admin.di.fc.ul.pt>

didatos fazem todos parte do público alvo desta empresa de modo a que todos possam interagir para efeitos de ensino ou aprendizagem.⁴

Actualmente a *Digitalis* serve um conjunto alargado de produtos à Universidade de Lisboa que se concentram na plataforma de gestão académica SiGES.

3.2 Plataformas

3.2.1 Moodle

Unidades Curriculares O Moodle organiza as unidades curriculares em *courses* aos quais associa os docentes e alunos. Com as configurações de permissões por omissão, apenas os docentes podem publicar os conteúdos e instanciar módulos para criar grupos de trabalho ou actividades, como sejam testes, entregas de trabalho, entre outros. As unidades curriculares do Moodle podem ter alunos associados para que estes possam aceder à informação e participar nas respectivas actividades. Cada unidade curricular no Moodle tem de pertencer a uma e só uma Categoria. Estas Categorias facilitam a navegação na busca da unidade curricular que se pretende aceder.

As categorias são o único mecanismo de organização de unidades curriculares do Moodle, não existindo outros conceitos que poderiam ser igualmente considerados úteis como o de ano lectivo ou semestre. Desta forma e atendendo à natural repetição de conteúdos entre instâncias da mesma unidade curricular em anos lectivos distintos, as Unidades de Informática de cada instituição com o apoio do *e-learning lab* repetem semestralmente um processo de preparação. Este processo exige que a plataforma Moodle esteja preparada para a publicação dos conteúdos, isto é, que as Unidades Curriculares estejam criadas e prontas para que os conteúdos possam ser inseridos.

Classificações No Moodle cada unidade curricular tem o seu livro de classificações onde aparecem todos os seus alunos e respectivas avaliações. É acompanhado dum relatório de classificações que colecciona informação de todas as actividades que dispõem de avaliação, feita através do Moodle ou divulgada pelo Docente, nessa unidade curricular. O livro de classificações do Moodle é dividido por componentes de avaliação e permite ainda a definição de regras complexas de ponderação dos elementos de avaliação no estabelecimento da classificação final. A exportação de notas no Moodle[9] pode ser feita pelos docentes da unidade curricular no formato Excel spreadsheet, OpenDocument spreadsheet, texto ou XML.

⁴<http://www.digitalis.pt>

3.2.2 SiGES

MOODLE-IS A informação proveniente do SiGES é sincronizada com as plataformas de apoio à aprendizagem através da publicação da informação de alunos, docentes e unidades curriculares por meio de serviços web. O MOODLE-IS é o módulo que concretiza esta publicação na plataforma Moodle, oferecendo às instituições da Universidade de Lisboa uma redução no fluxo de trabalho necessário para o processo de preparação do semestre lectivo, no que toca à criação de unidades curriculares que não existam no Moodle e à associação de alunos e docentes às unidades curriculares.

Sumários O SiGES permite a publicação dos conteúdos das aulas. A definição de sumário do SiGES segue os requisitos oficiais, contendo por isso um texto, dado por um Professor, associado a um número de presenças e a uma Aula. A aula é identificada por uma data e uma unidade curricular. Esta plataforma disponibiliza uma interface que é utilizada ao longo do semestre pelos docentes para a publicação dos sumários.

3.2.3 Requisitos funcionais

Os Serviços Centrais da Universidade de Lisboa, juntamente com o Departamento de Informática da Faculdade de Ciências da Universidade de Lisboa e com o e-learning Lab, organizaram um conjunto de requisitos funcionais que correspondem a problemas que têm vindo a ser identificados no contexto do funcionamento das plataformas de apoio à aprendizagem e de gestão académica.

Disponibilidade da plataforma Moodle No início do semestre, são preparados os conteúdos das unidades curriculares na plataforma de apoio à aprendizagem. Estes conteúdos podem ser novos ou aproveitados do ano lectivo anterior mas são sempre inseridos novamente na plataforma numa categoria dedicada ao semestre corrente. Este modelo tem levado a um aumento significativo do espaço ocupado em disco pela plataforma de semestre para semestre.

Numa fase inicial deste trabalho, acompanhando o início de um semestre, foram feitos alguns diagnósticos para avaliação dos recursos disponíveis e utilizados pela plataforma em produção. Durante este período observou-se que as actividades inerentes às Unidades Curriculares colocam a disponibilidade da plataforma em risco devido à lotação do espaço em disco do servidor em questão. Uma plataforma que está sempre a receber novos conteúdos e requer uma gestão complexa de informação de sessão, informação interactiva semi-permanente da qual depende a comunicação dos utilizadores, como é o caso do Moodle, necessita sempre de espaço para que esta esteja disponível. Se por alguma razão o servidor tiver o seu espaço em disco lotado ou tiver que processar sobre uma grande quantidade de dados, a disponibilidade da plataforma estará em causa e, por isso, houve

uma preocupação em libertar algum do espaço ocupado pelo Moodle, por via de instalação de várias instâncias desta plataforma e em desenhar um processo que permita prevenir incidentes.

Projecção de Unidades Curriculares entre Plataformas Toda a informação que diz respeito às unidades curriculares, seus docentes e inscrição de alunos está centralizada no SiGES. Actualmente essa informação, fundamental para o funcionamento adequado do Moodle, é exportada no início de cada semestre do SiGES para o Moodle num processo com intervenção humana. O MOODLE-IS tem como objectivo facilitar este processo de transferência de informação.

Contudo, esta solução ainda não é aplicável numa configuração onde existe mais que uma instância operacional de Moodle. É então necessário que o MOODLE-IS seja capaz de fazer a exportação dos dados para as unidades curriculares em diferentes plataformas Moodle.

De acordo com a análise feita na secção 3.2.2, uma vez que o MOODLE-IS segue um modelo para comunicação do SiGES com plataformas de apoio à aprendizagem, faz parte dos requisitos que a comunicação com base na nova configuração de várias instâncias seja também aplicada para qualquer dessas plataformas.

Apresentação e Organização das Unidades Curriculares Na Universidade de Lisboa a plataforma Moodle costuma ser organizada com a atribuição de uma categoria para cada unidade orgânica, dentro dessa unidade uma outra categoria para cada ano lectivo e, por fim, uma categoria para cada curso que contém as suas unidades curriculares.

Esta árvore de categorias resulta numa limitação identificada na Universidade de Lisboa quando uma unidade curricular é leccionada a múltiplos cursos. Pretende-se que uma mesma unidade curricular possa ser acedida por diferentes categorias, funcionalidade que o Moodle não oferece.

Projecção de Sumários entre Plataformas O sumário, que descreve o conteúdo duma aula, é uma das informações do interesse do Aluno que os Docentes costumam, por regra, partilhar através da plataforma de gestão académica. No entanto, como é um tipo de informação do interesse do aluno, deve também constar na plataforma de apoio à aprendizagem e, por isso, existe uma preocupação em torná-la acessível no Moodle.

É de certa forma dispendioso a colocação dos sumários das aulas por parte do docente em dois sistemas de informação distintos como sejam o sistema de informação académico e a plataforma de aprendizagem. Podemos identificar este esforço por observação da fig. 3.1 na presença de fluxos redundantes. A presença de quebras indica-nos os possíveis problemas que podemos ter com fluxos repetidos.

Pretende-se que os sumários sejam colocados apenas uma vez num dos sistemas e

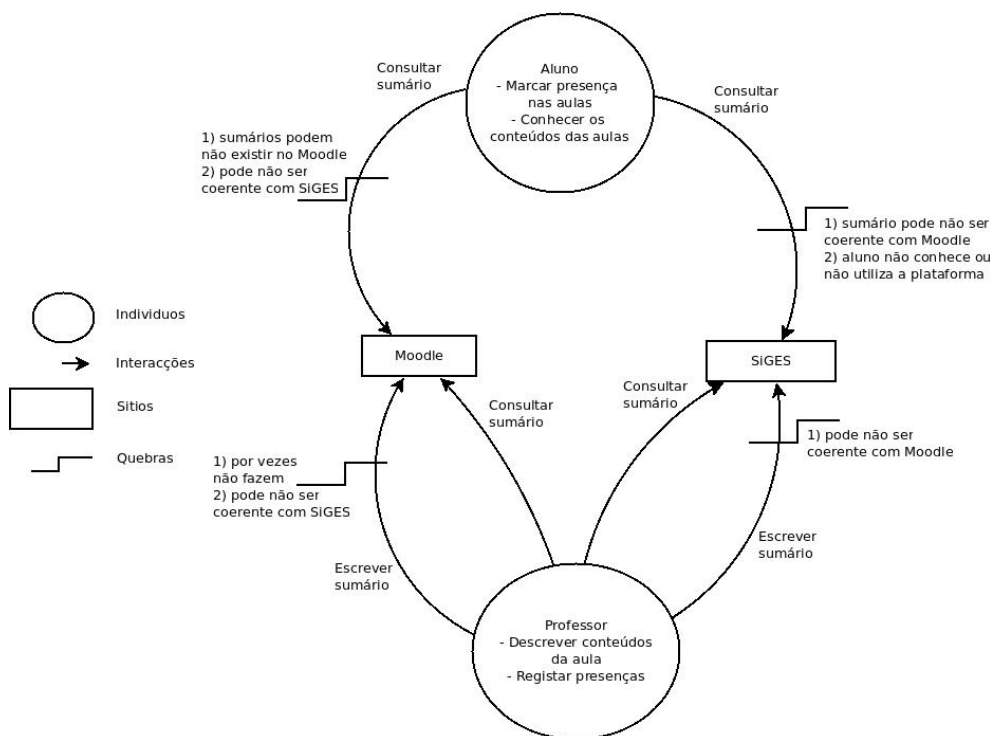


Figura 3.1: Mapa de Fluxo de Trabalho sobre Sumários

sejam acessíveis por ambos. De acordo com a proposta do novo mapa de fluxos de trabalho para os sumários (fig. 3.2) pretende-se criar um modelo bi-direccional onde o registo de informação é feito apenas no SiGES e acessível no Moodle, eliminando assim a redundância observada no mapa de fluxos original e reduzindo o seu número de quebras.

Projectão de Classificações entre Plataformas Se existisse uma ligação que permitisse projectar as componentes de avaliação dos cursos numa forma coerente entre os sistemas o docente poderia preencher a sua pauta de avaliação apenas uma vez, reduzindo o esforço actual de publicar as avaliações em ambas as plataformas, como ilustrado na fig. 3.3. Por análise desta figura podemos observar a presença de fluxos redundantes e a existência de quebras. Como tal, pretende-se que as avaliações possam ser comunicadas entre plataformas como identificado no diagrama da fig. 3.4, eliminando assim as quebras anteriormente identificadas.

No SiGES as componentes de avaliação podem ser diferentes das estabelecidas pelo docente no Moodle com as primeiras a apresentarem tipicamente uma granularidade maior, isto é, são componentes que abrangem um maior número de factores de avaliação. A projectão de classificações implicará uma comparação das componentes de avaliação de modo a que as primeiras sejam registadas nas componentes correctas antes do lançamento da pauta.

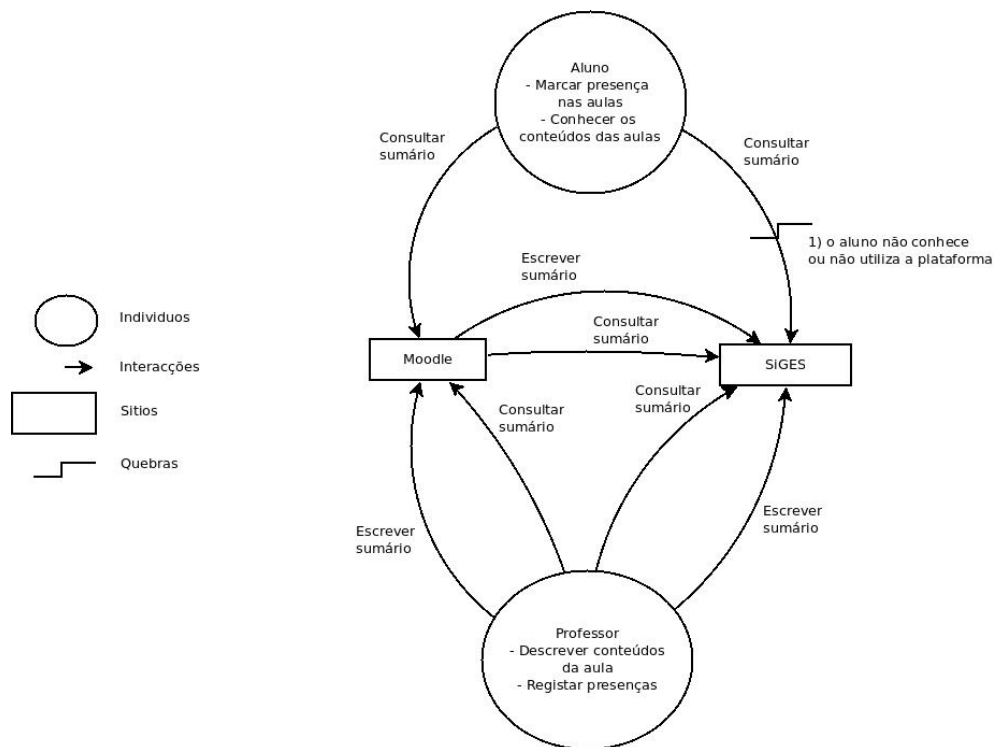


Figura 3.2: Mapa de Fluxo de Trabalho para solução proposta sobre Sumários

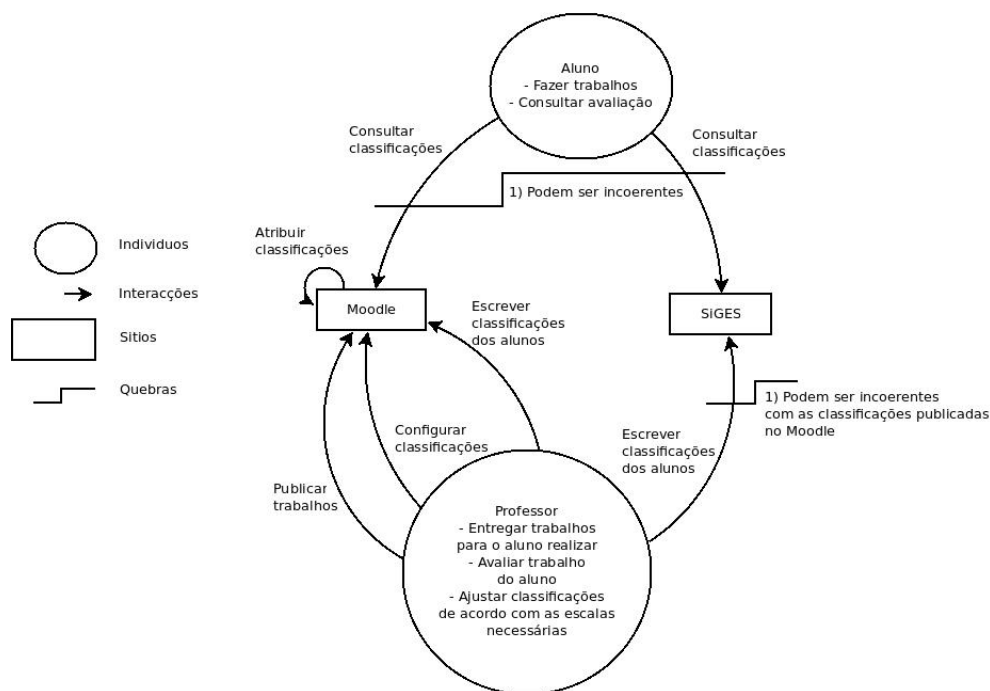


Figura 3.3: Mapa de Fluxo de Trabalho sobre Classificações

Módulos Moodle Faz parte dos requisitos deste trabalho que os módulos desenvolvidos sejam suficientemente normalizados de modo a que possam ser reutilizados em qualquer contexto da interação de outros sistemas com o Moodle por fim a oferecer uma maior

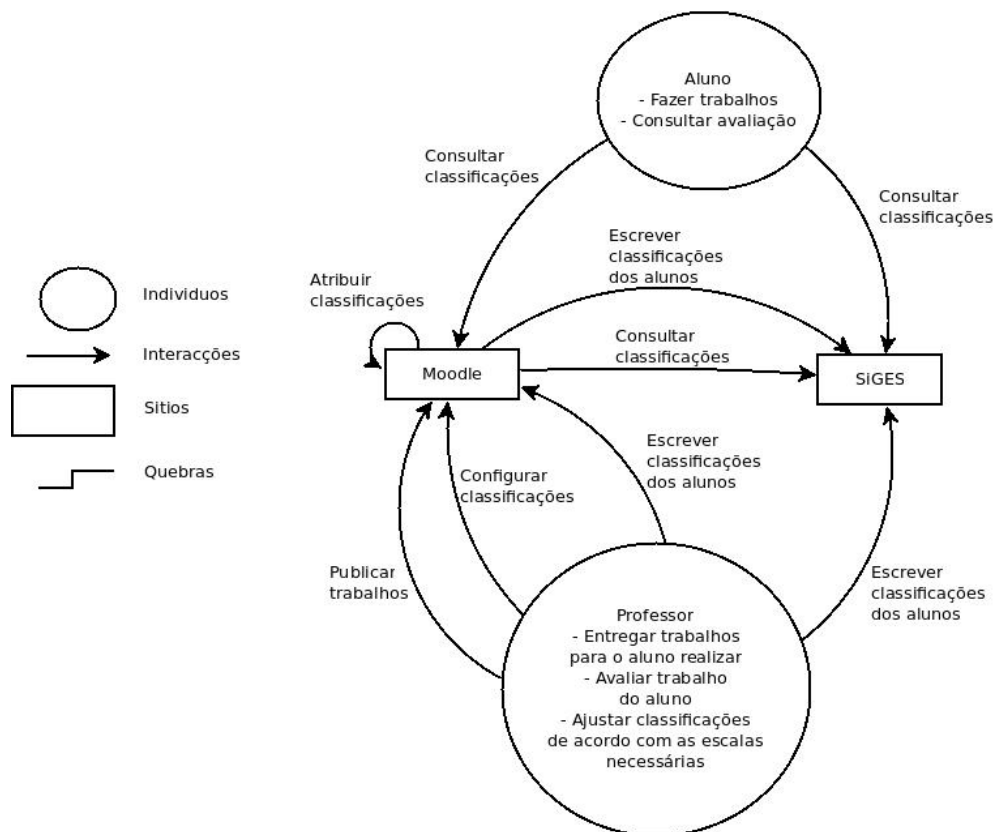


Figura 3.4: Mapa de Fluxo de Trabalho para solução proposta sobre Classificações

portabilidade do trabalho. Como tal, é proposto que os módulos sejam desenvolvidos com um nível de abstracção no seu modelo de comunicação por forma a que qualquer sistema possa vir a disponibilizar funcionalidades, à semelhança do SiGES.

Capítulo 4

Desenho

4.1 Histórico Moodle

Para atenuar o impacto provocado por uma dimensão excessiva da plataforma Moodle optou-se pela construção de um histórico para os conteúdos das unidades curriculares concluídas, isto é, foi instalada uma cópia da instância da plataforma em produção para a qual, num período de mudança de semestre, são copiados os conteúdos e organizados por ano lectivo à semelhança do que se faz no Departamento de Informática da Faculdade de Ciências.

Contudo, a instalação de um arquivo nas circunstâncias actuais obriga a um trabalho de transferência dos conteúdos que já não são necessários na plataforma em produção para a nova instalação. Avaliou-se este trabalho como moroso e justificativo de uma estratégia para diminuir o tempo despendido e os recursos humanos envolvidos.

Para a criação de um Histórico Moodle é necessária a instalação de uma nova plataforma seguida de um processo de migração de conteúdos da plataforma em produção para a nova. Por observação do caso prático do Departamento de Informática da Faculdade de Ciências da Universidade de Lisboa, assumiu-se que a instância de Histórico terá uma utilização muito inferior à instância de produção, ocupará menos recursos de processamento e memória volátil e tenderá a ocupar mais disco. Recorrendo às tecnologias de virtualização ao dispôr dos Serviços Centrais da Universidade de Lisboa instalou-se uma máquina virtual com Apache e MySQL e uma instalação Moodle.

Sobre a versão do Moodle para a instância do Histórico foi pensado um plano para uma instalação da versão 2.4 uma vez que a versão actualmente em produção 1.9 também seria actualizada para esta versão e, assim sendo, o histórico daria já o primeiro passo.

A migração consiste num processo de exportação de cada unidade curricular na instância de produção gerada através da funcionalidade de criação de cópias de segurança. Esta operação gera um ficheiro comprimido com toda a informação sobre a unidade curricular em questão. A instalação da unidade curricular no histórico é executada através de uma operação de restauro utilizando o ficheiro de salvaguarda na instância de Histórico. Du-

rante os testes desta operação detectámos a incompatibilidade de versões e percebemos que a sua resolução seria bastante morosa porque seria necessário executar um processo de conversão para cada ficheiro a restaurar. Foi então instalada uma versão 1.9 na instância do Histórico, igual à versão em produção e foi feito um plano para a redução do fluxo de trabalho na transferência de conteúdos recorrendo a scripts.

Num primeiro passo avaliou-se o processo de exportação de uma Unidade Curricular Moodle para o seu ficheiro comprimido. Por avaliação do funcionamento da plataforma entendeu-se que este executa um conjunto de cópias de segurança sobre todas as unidades curriculares cuja cópia está agendada. Contudo, pretende-se fazer esse conjunto de cópias sobre um contexto bem definido, isto é, um conjunto de unidades curriculares referentes a um ano lectivo. Por contexto da utilização da plataforma por parte do *e-learning lab*, entendemos que as Unidades Curriculares são inseridas numa categoria cujo nome tem identificado o ano lectivo e que, assim, é possível contextualizar as unidades curriculares que se pretendem executar a cópia de segurança.

Então, num segundo passo foi desenhado um processo, como ilustrado na fig. 4.1), com recurso a um novo script que permite a preparação do conjunto de unidades curriculares cuja cópia de segurança é necessária e que permite executar processo através da linha de comandos.

Num terceiro passo avaliou-se a localização destes ficheiros comprimidos e concluiu-se que depende da configuração das cópias de segurança na plataforma. Conhecendo o local onde os ficheiros são guardados preparou-se um conjunto de comandos responsável pela transferência dos ficheiros comprimidos entre máquinas. Recorrendo ao rsync[17], ferramenta que permite uma transferência de ficheiros com avaliação de coerência entre duas localizações, preparou-se então o script de transferência dos ficheiros da instância em produção para a instância de histórico.

Num último passo avaliou-se o processo de restauro das cópias de segurança do Moodle na versão 1.9 e concluiu-se que automatizar este processo seria complicado devido às fases de configuração que são necessárias executar. Quando é iniciado um processo de restauro é necessário identificar a colocação da unidade curricular no que diz respeito à categoria que a irá conter e à informação que se pretende importar. Sendo assim, averiguou-se um modo de preparar os ficheiros todos numa pasta acessível pelo administrador que é responsável por aceder ao ficheiro através da interface da plataforma e executar o processo de restauro.

4.2 Apresentação e Organização das Unidades Curriculares

Como identificado anteriormente, o Moodle disponibiliza as categorias como método de organização das unidades curriculares. O modelo de dados de categorias no Mo-

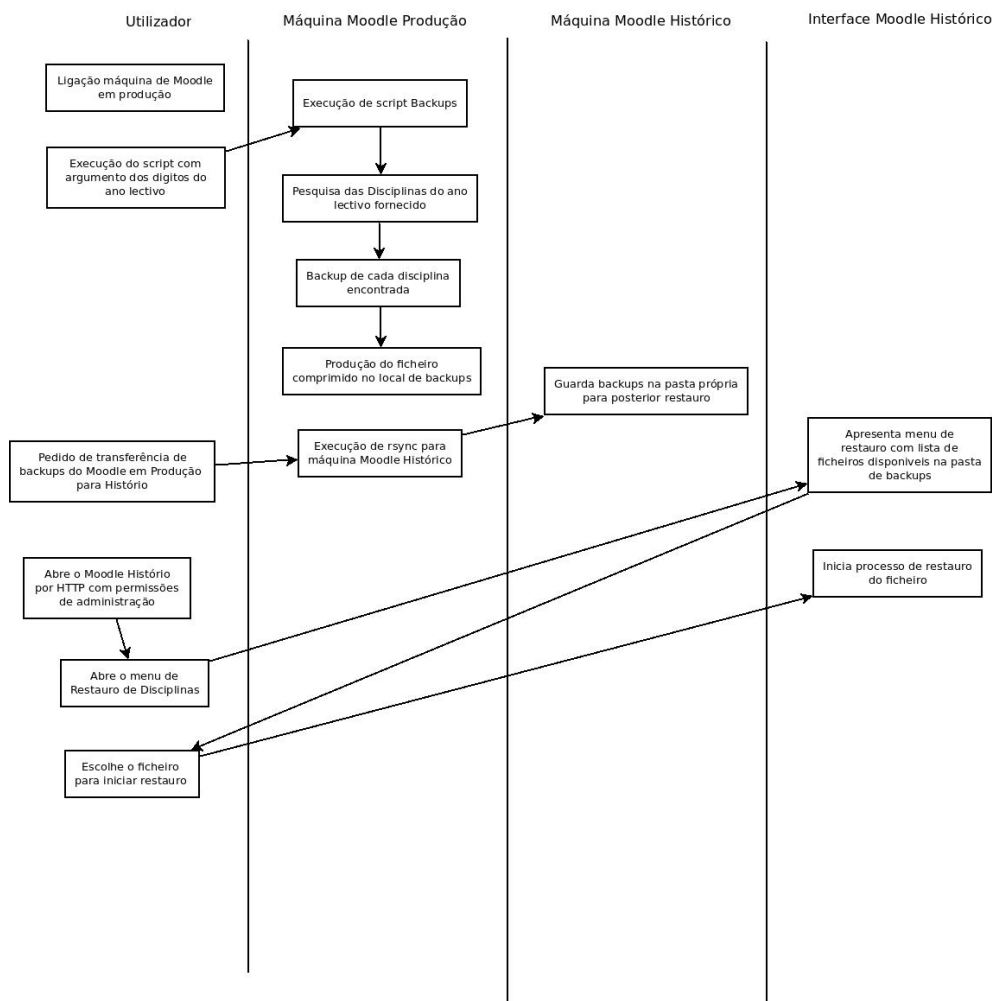


Figura 4.1: Diagrama de Operações para Transferência de Disciplinas entre instâncias Moodle

odle, como ilustrado na fig. 4.2, contém um conjunto de atributos que permite a sua identificação por um número único, um nome e um outro número de ordem.

A equipa de Administração de Sistemas do Departamento de Informática da Faculdade de Ciências já tinha identificado o requisito da apresentação e organização das unidades curriculares cuja solução foi conseguida com um novo desenho da estrutura de dados como apresentado no diagrama da fig. 4.3. Por análise deste diagrama podemos observar que a relação entre uma Unidade Curricular e o elemento principal da sua organização, a categoria, foi alterada por fim a conseguir um grau de multiplicidade de muitos para muitos. Para que esta alteração fosse possível sem alterar o modelo original foram criadas novas tabelas de acordo com este novo modelo que serão utilizadas quando o novo esquema de apresentação é desejado. A categoria denomina-se neste módulo por *Degree*.

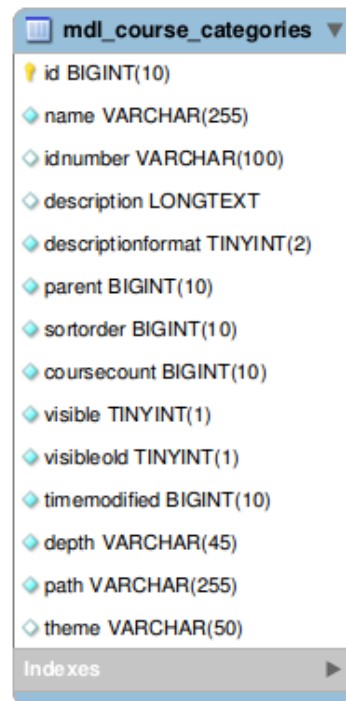


Figura 4.2: Modelo de Dados de Categorias no Moodle

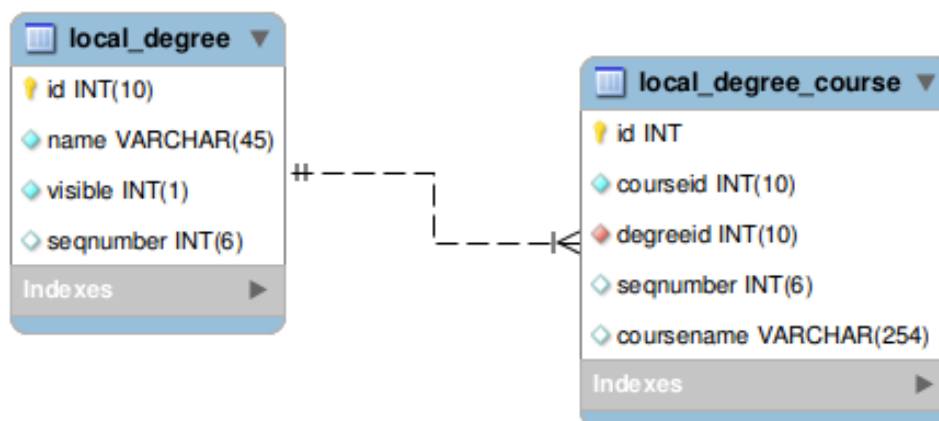


Figura 4.3: Modelo de Dados do Módulo para Categorias

4.3 MOODLE-IS com Projecção Multi-instâncias

O MOODLE-IS está preparado para comunicar com o Moodle, como já mencionado, com o objectivo de projectar nesta plataforma de apoio à aprendizagem as respectivas unidades curriculares. Para tal, nas suas configurações são guardados dados para a comunicação com o serviço web do Moodle e alguns parâmetros que definem o processo respectivo à projecção de informação. Com as alterações propostas pretende-se que este módulo comunique com mais do que uma instância Moodle e, por isso, detenha dados de configuração

para mais que um serviço web respectivo a cada plataforma de aprendizagem.

4.3.1 Ponto de situação

Na versão em produção deste módulo é possível um funcionário executar uma projecção de unidades curriculares do SiGES no Moodle por via de um conjunto de passos de configurações e execuções como podemos observar na fig. 4.4. Num cenário de multi-instância a projecção de uma unidade curricular deve ser feita numa das instâncias Moodle escolhidas à priori. A escolha desta instância, onde se quer projectar a unidade curricular, deve ser feita de acordo com o ano lectivo a que esta corresponde. Sendo assim, num cenário multi-instância o processo de projecção (fig. 4.4) exige que seja acrescentado um primeiro passo onde é feita uma escolha do ano lectivo que se quer projectar. A nova versão é apresentada no diagrama da fig. 4.5.

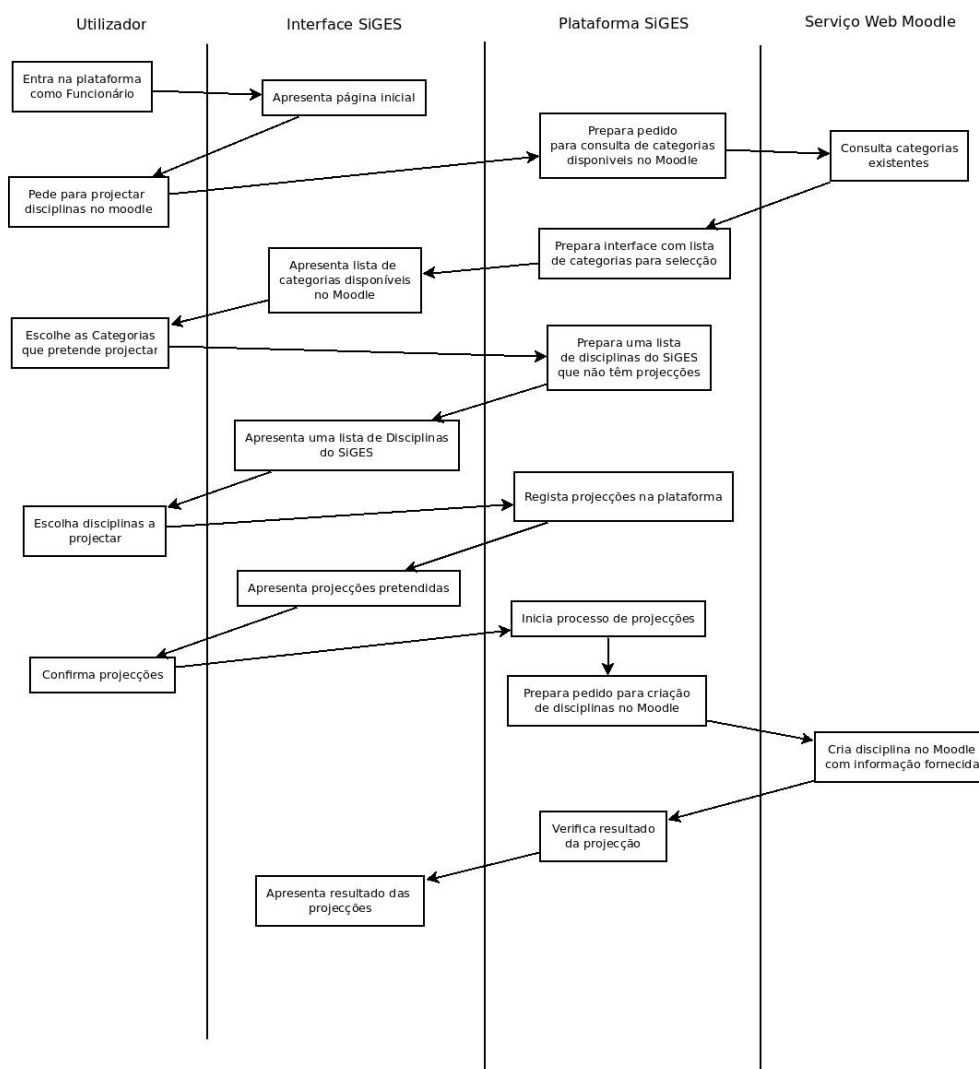


Figura 4.4: Diagrama de Operações para Projectar Disciplinas SiGES no MOODLE

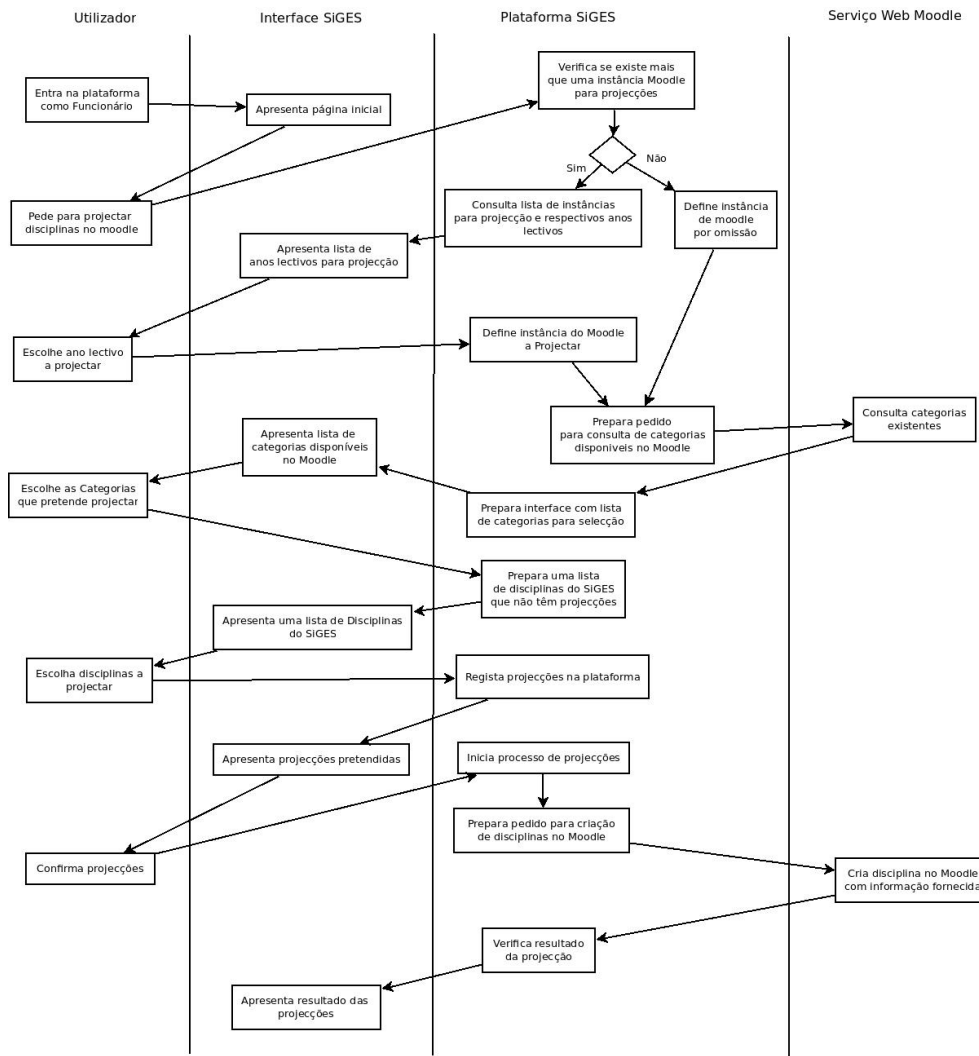


Figura 4.5: Diagrama de Operações para Projectar Disciplinas SiGES em várias plataformas Moodle

Foi feito um levantamento do funcionamento do MOODLE-IS no que diz respeito aos elementos necessários para a configuração da comunicação com o Moodle. Dentro do projecto Maven *dif-elearning* existe um conjunto de classes Java responsáveis pela comunicação. Como ilustrado no Diagrama de Classes da fig. 4.6 podemos observar a existência de uma classe *MoodleConfiguration* cujos atributos definem as configurações necessárias à comunicação e ao processo de negócio executado pelo MOODLE-IS. Esta configuração é utilizada por uma outra classe de nome *ELearningMoodleImpl* que é responsável pela concretização da comunicação com a plataforma em questão.

4.3.2 Solução proposta

Com o objectivo de concretizar uma relação de multiplicidade entre as duas classes ilustradas na fig. 4.6 foi concretizada uma nova classe de nome *ELearningInstanceManager*.

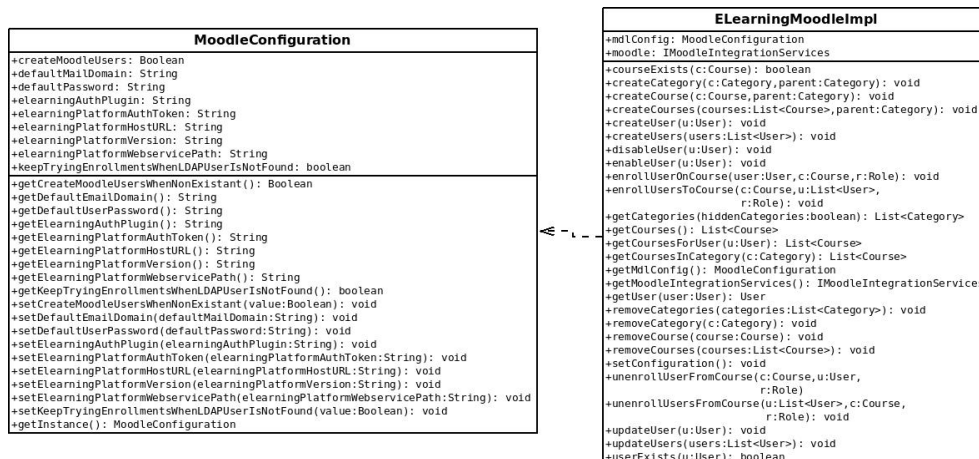


Figura 4.6: Parte do Diagrama de Classes dif-elearning

Como o nome indica, esta classe detém a informação de negócio necessária para gerir um cenário de multi-instâncias Moodle e será responsável por gerir os vários objectos *ELearningMoodleImpl* necessários para comunicar com as várias plataformas.

Como ilustrado no Diagrama da fig. 4.7 a classe *ELearningMoodleImpl* dispõe de métodos para gerir as várias instâncias, para consultar elementos de configuração de uma determinada instância pela sua chave de negócio e consultar as chaves de negócio disponíveis, como por exemplo os anos lectivos cujas projecções de unidades curriculares estão activas. Com o objectivo de não deixar de considerar o cenário anterior, em que apenas existe projecção para uma instância Moodle, foi utilizada a classe *ELearningConfiguration*, responsável pela configuração do módulo *dif-elearning* onde é guardada informação das projecções existentes e onde pode ser verificado se o cenário é de multi-instância ou não. Sendo assim, a classe *ElearningInstanceManager* será o novo ponto de entrada quando o MOODLE-IS necessita de comunicar com uma instância Moodle e, por isso, este módulo terá que ser alterado em qualquer sítio em que é necessária a utilização do objecto *ELearningMoodleImpl*.

4.4 Comunicação entre plataformas

4.4.1 Modelo de Comunicação

Recorrendo, então, à tecnologia de serviços web, foi feita uma análise sobre a arquitectura a utilizar para a comunicação entre plataformas. Nesta análise foram consideradas as duas arquitecturas SOAP e REST. Atendendo os requisitos referidos no capítulo anterior, uma comunicação bem estabelecida e escalável nos recursos a disponibilizar pelo servidor e a capacidade do SOAP para definir um contracto de comunicação optou-se pela utilização deste protocolo.

Num primeiro passo concretizou-se um desenho do contracto da comunicação dos

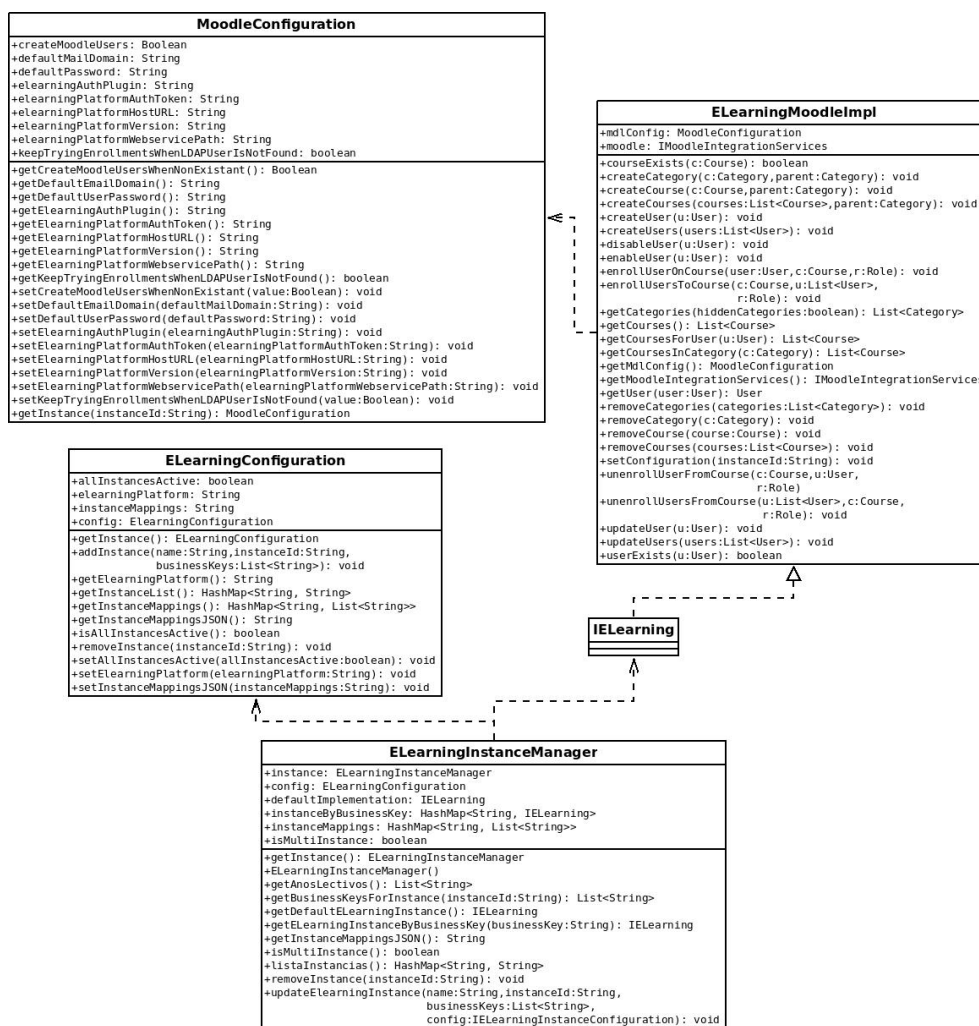


Figura 4.7: Alterações ao Diagrama de Classes dif-elearning

serviços web necessários neste trabalho. Foi feito um desenho do fluxo das operações a serem concretizadas pelo utilizador, das funções necessárias e dos dados a comunicar entre as plataformas.

4.4.2 Comunicação de sumários entre plataformas

Neste contexto dois cenários são aplicados: O utilizador pode aceder ao Moodle para consultar os sumários de uma certa unidade curricular como ilustrado no diagrama da fig. 4.8 ou pode utilizar a mesma plataforma para enviar os sumários para o SiGES como ilustrado no diagrama da fig. 4.9.

Pela análise concretizada no capítulo anterior, podemos constatar que um sumário descreve o conteúdo duma aula num determinado momento da unidade curricular. Com o objectivo de abstrair esta informação assumimos que desconhecemos o contexto dos dados. Então, para a escrita dum sumário o utilizador necessita conhecer à priori os elementos que o representam, por exemplo, as turmas que lecciona duma certa unidade

curricular. Esta é a informação necessária a encapsular num conjunto de mensagens a comunicar com o serviço como ilustrado na fig. 4.11 e de acordo com o processo de comunicação identificado no diagrama da fig. 4.10. Existe uma função para a leitura das componentes de sumário dada uma unidade curricular, provenientes da plataforma de gestão académica, uma função de escrita de um sumário com base nas componentes fornecidas e por último, para consulta, existe uma terceira função que permite uma leitura de um conjunto de sumários para uma determinada unidade curricular. Assim ficou definido um serviço a que chamámos de SummaryWS.

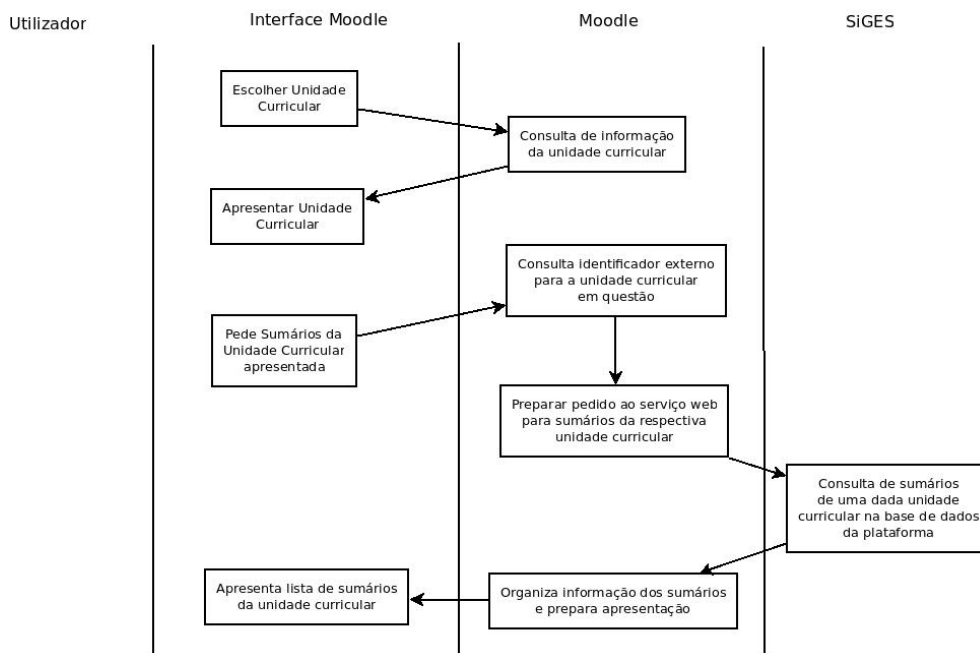


Figura 4.8: Fluxo de operações para apresentação de Sumários

4.4.3 Comunicação de classificações entre plataformas

No contexto da comunicação de uma pauta de avaliações numa unidade curricular do Moodle com o SiGES o utilizador, à semelhança dos sumários, necessita conhecer à priori as componentes de avaliação disponíveis na plataforma de gestão académica por forma a inserir as avaliações nas respectivas componentes, como identificado no diagrama da fig. 4.12. Estas componentes necessitam de ser validadas no que diz respeito à informação que as definem, isto é, o intervalo das suas notas mínima e máxima devem ser comparadas pois a escala de avaliação das componentes deve ser idêntica. Para efeito de comunicação, como ilustrado no diagrama da fig. 4.13, é necessária uma função para consulta dessas componentes de avaliação, uma outra função para a escrita de uma avaliação na respectiva componente e uma função para consulta das avaliações projectadas na plataforma de gestão académica. Pela figura 4.14 podemos observar a informação necessária para a

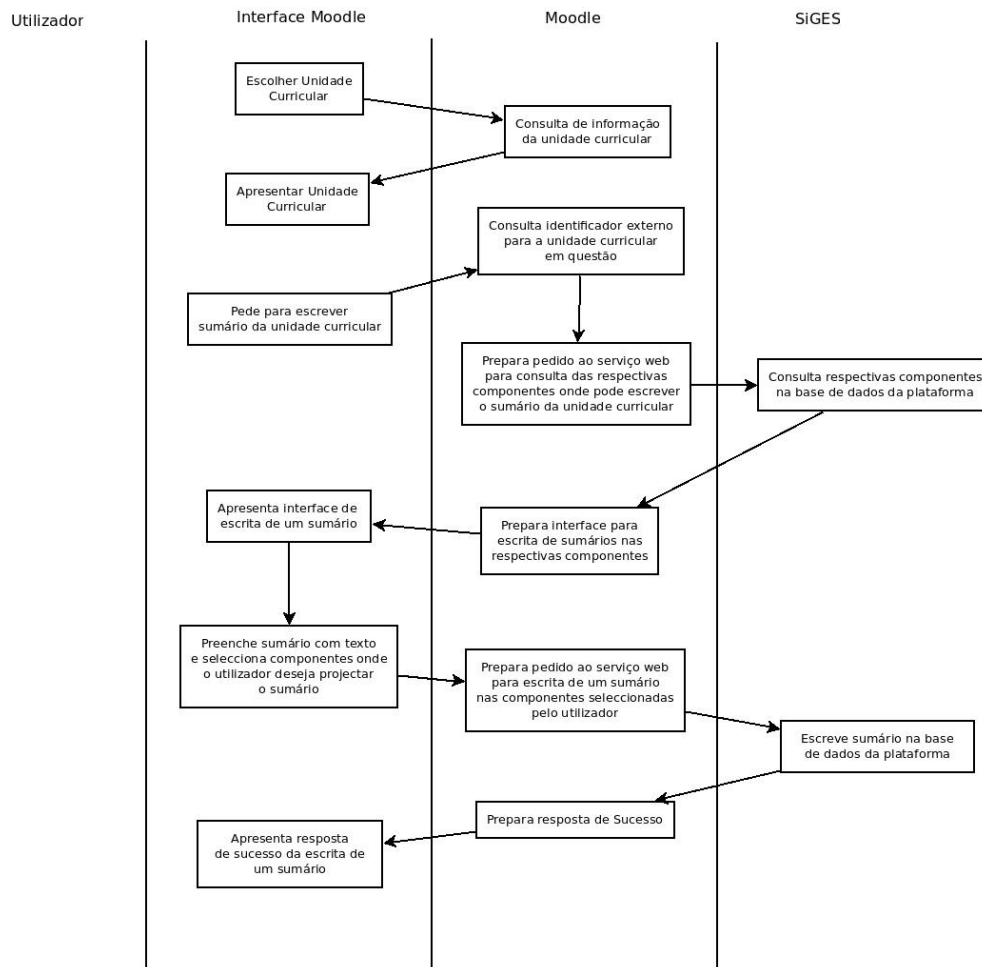


Figura 4.9: Fluxo de operações para escrita de Sumários

comunicação encapsulada num conjunto de mensagens. Assim ficou definido um serviço a que chamámos de GraderWS.

4.4.4 Dados a guardar no serviço web

Por forma a testar o contracto de comunicação foi desenhado um serviço web de testes que implementa o ponto de entrada do contracto de comunicação, que acede a uma base de dados cujo modelo da fig. 4.15 foi desenhado em concordância com a análise de requisitos feita neste trabalho e onde é simulada a projecção de informação da plataforma de gestão académica.

4.4.5 Desenho do cliente do serviço web

Sendo o cliente do serviço web uma plataforma em PHP foi então desenhado um cliente SOAP nesta linguagem para cada um dos serviços GraderWS e SummaryWS que comunicam segundo um contracto e que preparam os objectos das respostas para serem

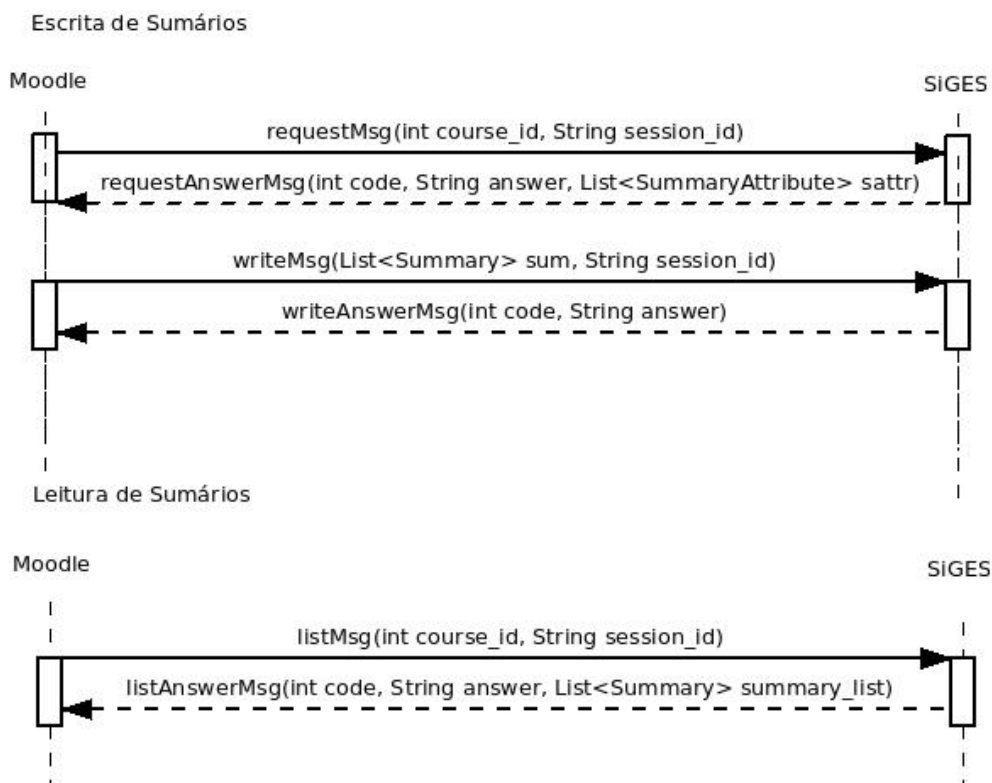


Figura 4.10: Comunicação de sumários com Serviço Web

posteriormente tratados pelos respectivos Módulos. Como ilustrado na fig. 4.16 foram desenhadas duas classes PHP com as funções necessárias para a comunicação com o serviço web, um cliente SOAP PHP responsável pelo encapsulamento das mensagens SOAP e um objecto de configuração que permite activar ou desactivar o modo de depuração de erros da comunicação.

4.5 Interfaces dos módulos Moodle

Com o objectivo de preparar o desenvolvimento dos módulos a implementar no Moodle foram concretizados protótipos de alta fidelidade recorrendo à API[12] que o Moodle disponibiliza para a sua interface.

4.5.1 Módulo Degrees

Foram desenhadas novas interfaces de utilização para a criação e edição dos Degrees e para a organização das respectivas unidades curriculares. Para adicionar um novo Degree, como podemos observar na fig. 4.17, é necessário o preenchimento de um formulário com um nome que a identifica e um número para ordenação. Com o fim de projectar uma unidade curricular nesse degree é necessária a indicação da unidade curricular que queremos projectar, um nome alternativo (opcional) para apresentação e um número para

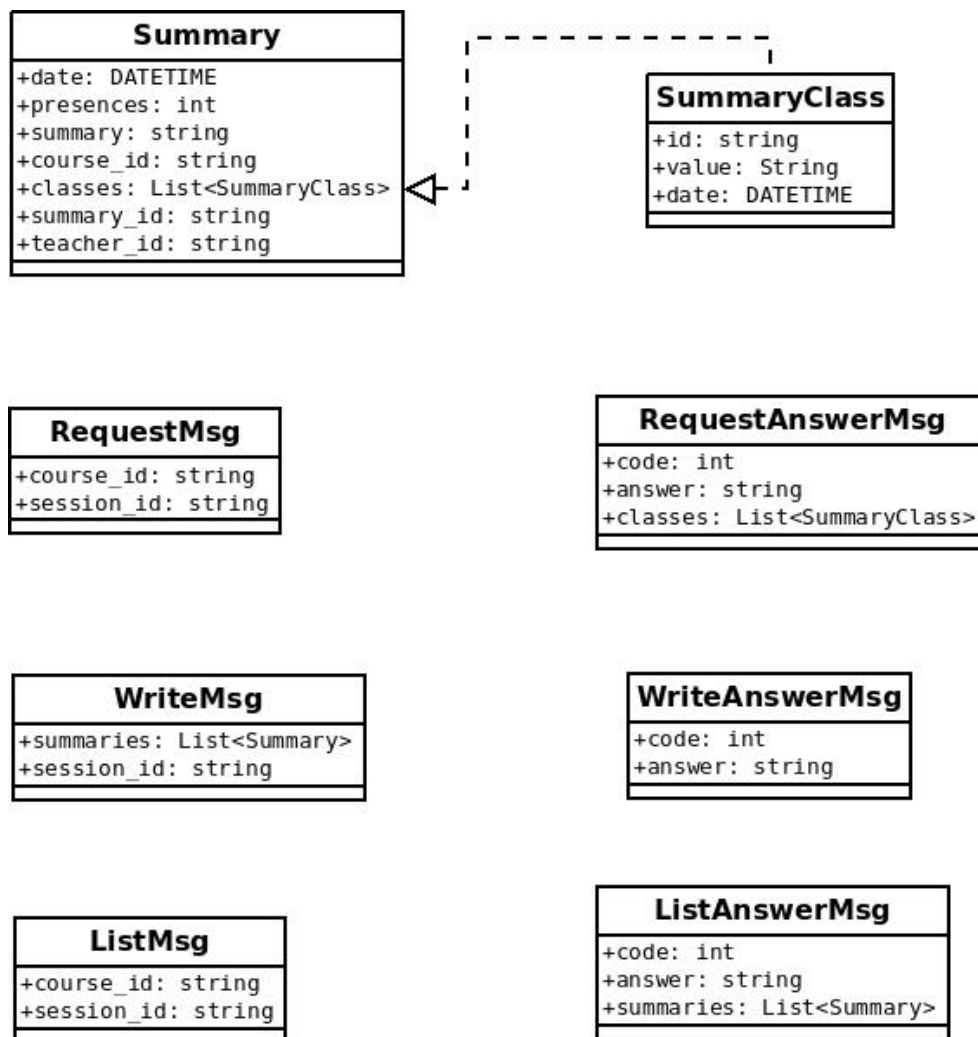


Figura 4.11: Mensagens necessárias para comunicação de Sumários com Serviço Web

ordenação como apresentado na fig. 4.18.

Para apresentar os degrees existentes foi criada uma tabela com os nomes que os identificam dispostos como representado na fig. 4.19. Depois de seleccionado um degree são apresentadas as correspondentes unidades curriculares devidamente ordenadas como representado na fig. 4.20. Esta última interface de utilização dispõe também de ligações para edição do degree em questão e para projecção de novas unidades curriculares.

4.5.2 Módulo SummaryWS

Atendendo aos requisitos de criação e consulta de sumários pela plataforma Moodle foram desenhadas novas interfaces de utilização. Para a criação de novos sumários, como representado na fig. 4.21, foi desenhado um formulário com um editor de texto, uma caixa para inserir o número de presenças e uma lista de possíveis componentes sobre as quais o docente pode escrever o sumário. Para a interface de apresentação dos sumários de uma

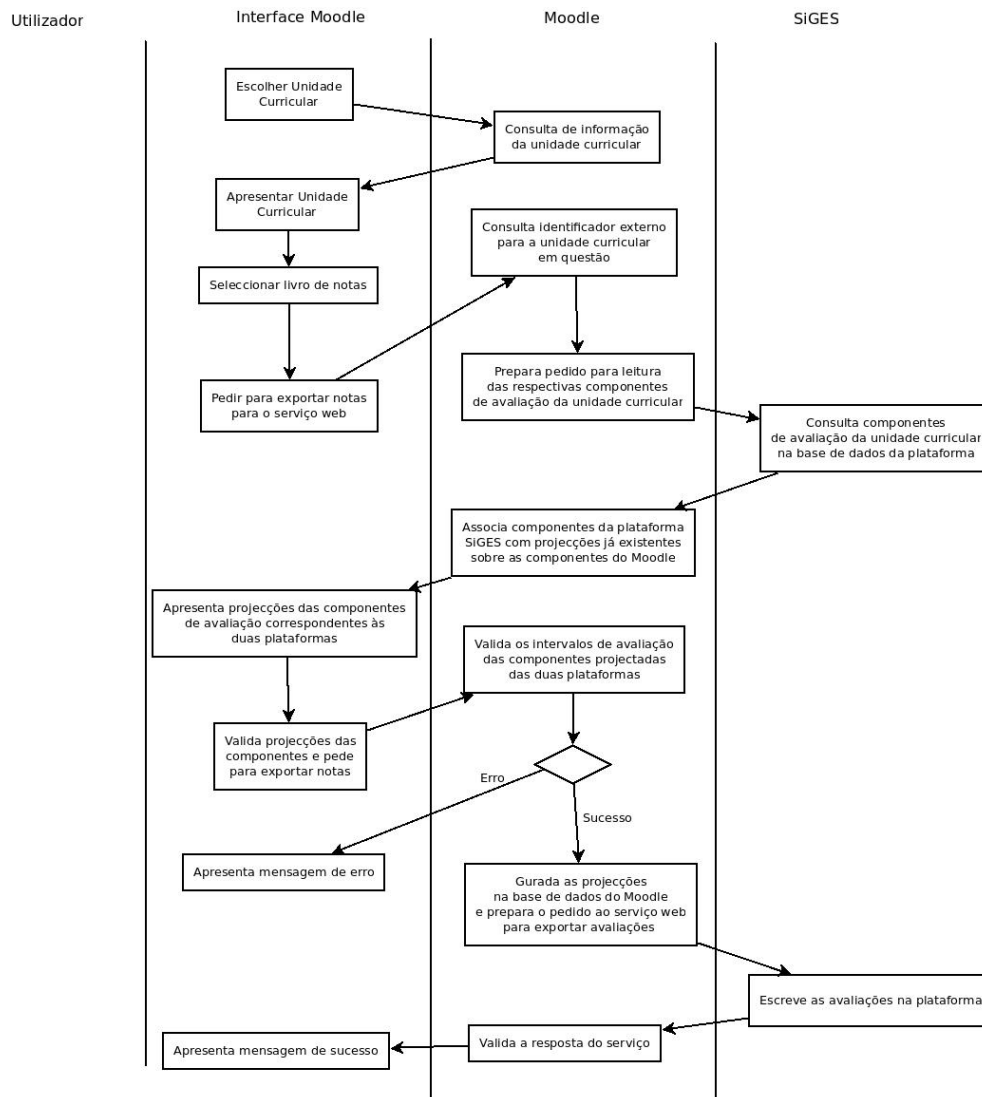


Figura 4.12: Fluxo de operações para exportação de Classificações

determinada unidade curricular foi desenhada uma tabela com um sumário por linha e a respectiva informação distribuída por colunas como representado na fig. 4.22.

4.5.3 Módulo GraderWS

Na projecção das classificações foi utilizada a interface de exportação já existente no Moodle. O problema das diferentes granularidades das componentes de avaliação é delegado no docente que atribui através da interface cada componente de avaliação do Moodle a uma componente de avaliação no SiGES. Como podemos observar pela fig. 4.23, foi organizada uma tabela onde na primeira coluna são apresentadas as componentes de avaliação referentes à unidade curricular em questão e na segunda coluna as possíveis componentes de avaliação para preenchimento no SiGES.

Ao submeter as projecções o docente entra numa segunda interface que indica o es-

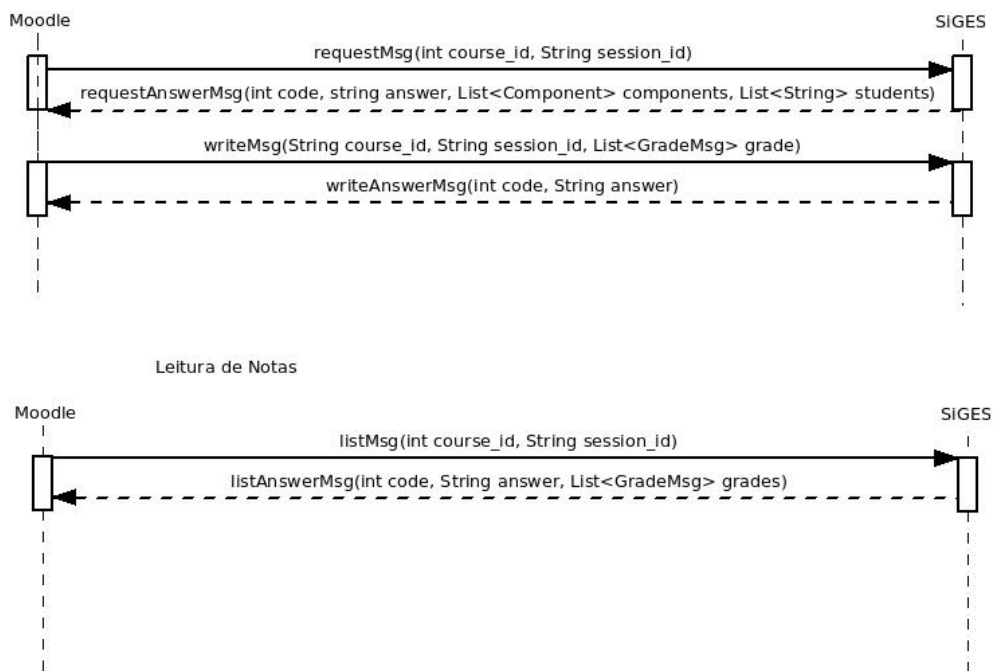


Figura 4.13: Comunicação de Classificações com Serviço Web

tado das mesmas. Sendo assim pode ser apresentada uma interface onde é indicado que as projecções estão correctas, como representado na fig. 4.24, ou incorrectas, como representado na fig. 4.25.

Caso as projecções estejam correctas o docente pode iniciar o processo de exportação das avaliações pelo que será apresentada uma última interface onde mostra uma tabela organizada com um aluno por linha e com as classificações das componentes de avaliação distribuídas nas colunas, como representado na fig. 4.26. Esta apresentação serve para o docente verificar como ficaram as projecções de classificações no SiGES.

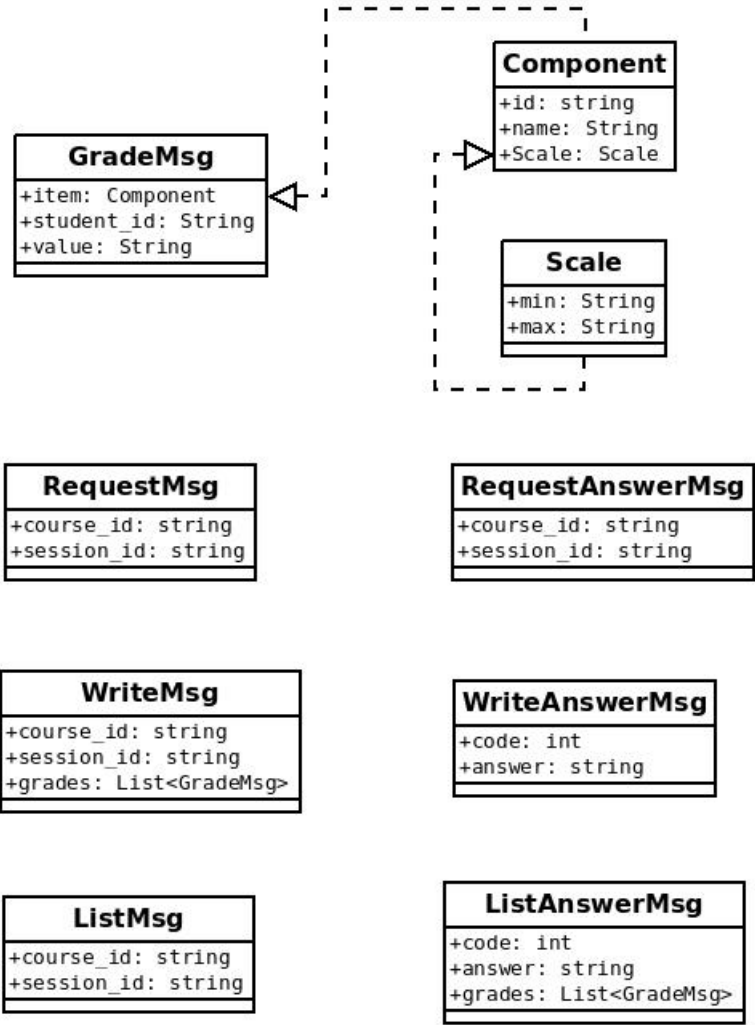


Figura 4.14: Mensagens necessárias para comunicação de Classificações com Serviço Web

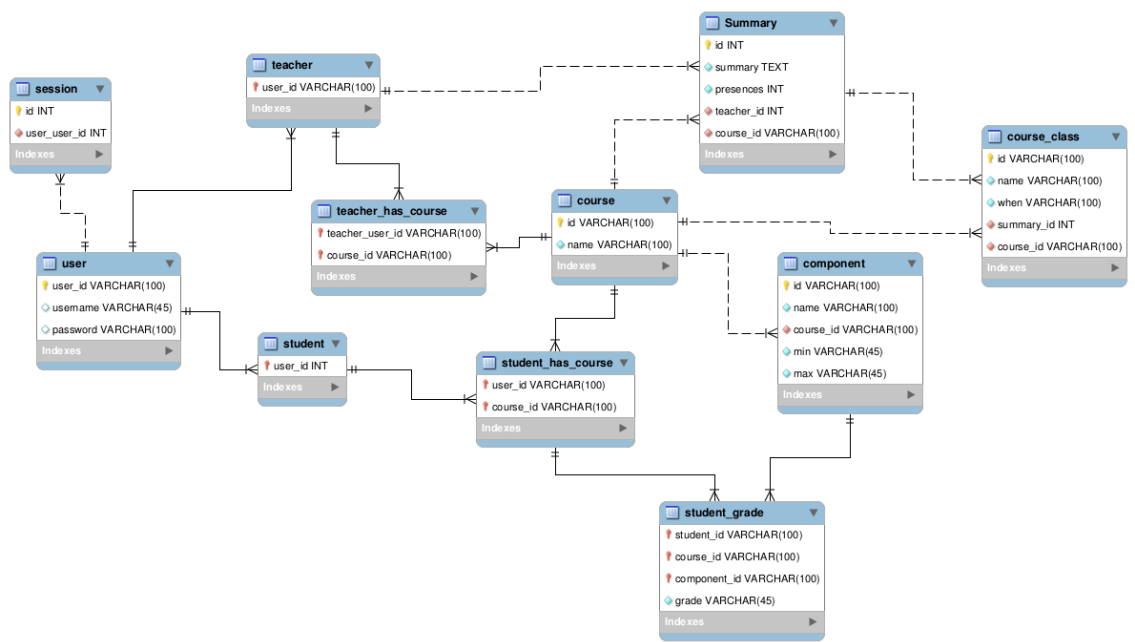


Figura 4.15: Modelo de Dados do Serviço Web

GraderWS
+client: SoapClient +session_id: String +config: Array
+__construct(): GraderWS +request(course_id:String): Array +write(course_id:String,component_id:String, student_id:String,student_grade:String): Array +listt(course_id:String): Array +print_trace()

SummaryWS
+client: SoapClient +session_id: String +config: Array
+__construct(url:String): SummaryWS +write(course_id:String,teacher_id:String, presences:int,text:String,Classes:Array, summary_id:String=null): Array +listt(course_id:String) +print_trace()

Figura 4.16: Comunicação de Sumários

Add new category

Category name*

Sequence number

Create categoryCancel

There are required fields in this form marked *.

Figura 4.17: Criação de uma Categoria

Add Course to Degree

Select course

Um curso exemplo

Course name in this Degree

Sequence number

Save changesCancel

Figura 4.18: Projecção de uma Unidade Curricular numa Categoria

Course categories

Um degree exemplo

Um segundo Degree exemplo

Figura 4.19: Lista de Degrees

Course categories: Um degree exemplo

Edit DegreeAdd Degree CourseDisable edit mode

CoursesEditSelect

Um curso exemplo com outro nome

Um segundo curso exemplo

Re-sort courses by nameAdd a new course

Search courses:Go

Figura 4.20: Unidades Curriculares de um Degree

Summary

Sumário	Turmas	Professor	Presenças	Admin
Um sumário	turma 1 em Thursday, 1 January 1970, 12:33 AM turma 2 em Thursday, 1 January 1970, 12:33 AM	Francisco Soares	10	Edit
um sumário diferente	turma 3 em Thursday, 1 January 1970, 12:33 AM	Francisco Soares	3	Edit

Figura 4.21: Lista de Sumários

Send new Summary

Lesson Summary

Font family ▼
Font size ▼
Paragraph ▼

B *I* U ABC X₂ X²

HTML

Um sumário

Path: p

Class presences

turma 1 (Thursday, 1 January 1970, 12:33 AM)

☒

turma 2 (Thursday, 1 January 1970, 12:33 AM)

☒

turma 3 (Thursday, 1 January 1970, 12:33 AM)

☐

Send Summary

Cancel


Figura 4.22: Escrita de um Sumário

Export to Grader Web Service

Options

Include feedback in export

☐

Require active enrolment 

☐

Preview rows

10

▼

Export new or updated grades only

☐

Grade export display type

Real

▼

Grade export decimal points

2

▼

Grade items to be included

Nota de Projecto (t1)

Nota final Projecto

▼

Nota final de Teste (t2)

Teste

▼

Final

Nenhuma

▼

Submit

Figura 4.23: Interface para Configuração de Projecções das Componentes de Avaliação



Figura 4.24: Mensagem de Sucesso da avaliação das Projecções de Classificações

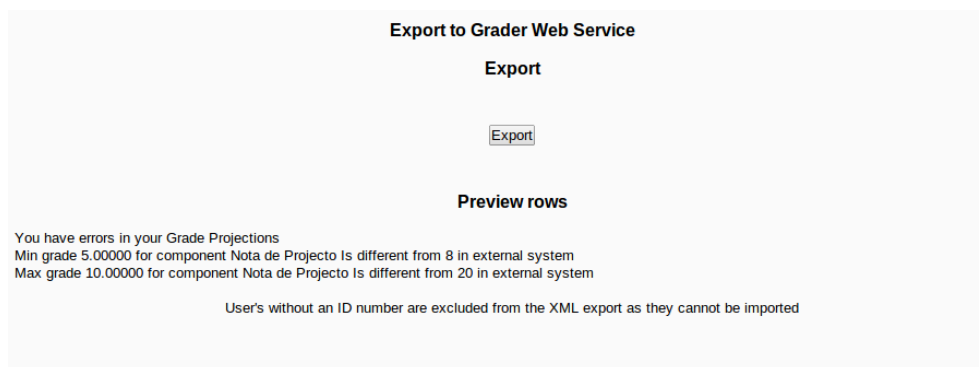


Figura 4.25: Mensagem de Erro da avaliação das Projecções de Classificações

Grader Web Service			
	Aluno	Nota Final Projecto	Teste
Francisco Soares		20.00	15.00

Figura 4.26: Resultado do Processo de Exportação de Classificações

Capítulo 5

Desenvolvimento

5.1 Histórico Moodle

Cópias de Segurança das Unidades Curriculares O Moodle dispõe de um script PHP que é executado periodicamente[7]. Este script é utilizado para a realização das cópias de segurança das Unidades Curriculares¹. Com o objectivo de executar um conjunto de backups das Unidades Curriculares, dado um contexto e poupando o esforço do uso da interface para fazer um backup de cada vez, foi feita uma cópia desse script e algumas alterações ao respectivo.

Num primeiro passo é necessário definir o contexto das Unidades Curriculares dados os argumentos de entrada, isto é, um conjunto de argumentos introduzidos pela linha de comandos que definem os anos lectivos sobre os quais se pretendem executar cópias de segurança. Num segundo passo é necessário executar as cópias de todas essas unidades curriculares e arrumar os ficheiros num sítio acessível para, num terceiro passo, executar uma transferência dos ficheiros para a instância do histórico. Por fim, é necessário executar o restauro dos ficheiros nessa instância.

Para a introdução de argumentos no contexto, sabendo que as Unidades Curriculares do Moodle da Universidade de Lisboa estão contidas numa categoria cujo nome tem o respectivo ano lectivo, foi concretizado um script de nome *mdlcategoriesano.php* que, como ilustrado na fig. 5.1, executa uma chamada MySQL à base de dados do Moodle onde o argumento de pesquisa é o ano lectivo e a resposta é um conjunto de categorias Moodle que contêm unidades curriculares referentes a esse ano lectivo.

Para a execução das cópias de segurança foi feita a cópia do script, já existente no Moodle, *cron.php* e foi alterada para aceitar argumentos pela linha de comandos[15], ou seja uma lista de categorias sobre a qual se pretende obter as unidades curriculares. O novo script foi chamado *admindb_backup.php*. A este script foi acrescentada uma biblioteca de funções auxiliares, denominada *admindb_lib.php* e ilustrada na fig. 5.4, que permite pesquisar todas as unidades curriculares referentes às categorias introduzidas. Foi também

¹https://docs.moodle.org/24/en/Backup_and_restore_FAQ

aproveitado o código de execução de cópias de segurança e foi removido todo o código que executa operações que não são necessárias. Na figura 5.2 podemos ver a introdução do script que foi acrescentada. Uma vez instalados os scripts *adminindi_backup.php* e *adminindi_lib.php* na pasta *admin* da plataforma Moodle, este processo pode ser executado a partir da linha de comandos como ilustrado na fig. 5.3 na presença do script *mdlcategoriesano.php*.

```
<?php
//Config CL002
$db_username = " ";
$db_password = " ";
$db = "db_elearning";

$user_query = "SELECT mdl_course_categories.id as id from mdl_course_categories
join mdl_course on mdl_course_categories.id = mdl_course.category where
mdl_course_categories.name like '%".$argv[1]."' group by id";

$link = mysqli_connect("localhost", $db_username, $db_password, $db);

if (mysqli_connect_errno()) {
    printf("Connect failed: %s\n", mysqli_connect_error());
    exit();
}

if ($result = mysqli_query($link, $user_query)) {
    while($results = mysqli_fetch_array($result)) {
        print($results['id']."\n");
    }
}

mysqli_close($link);

?>
```

Figura 5.1: Script de Contexto para o Backup de Unidades Curriculares

```
$numargs = count($argv);
$arg_list = $argv;
$categories = array();
for ($i = 0; $i < $numargs; $i++) {
    $categories[] = $arg_list[$i];
}
$categories_list = array();
foreach($categories as $id) {
    $categories_list[] = $id;
    $childs = childs($id,$link);
    foreach ($childs as $chid) {
        $categories_list[] = $chid;
    }
}
$courses = courses($link,$categories_list);
```

Figura 5.2: Introdução do Script de Backup

Uma vez produzidos os ficheiros de backups estes devem ser transferidos para a segunda instância Moodle. Num primeiro passo deve ser configurada uma pasta na segunda

```
#!/bin/sh
php mdlcategoriesano.php $1 > logs/categories$1.log
while read line
do
    cat=$line
    ./backup_category.sh $cat > logs/cat/cat-$cat.log
done < logs/categories$1.log
```

Figura 5.3: Script para Execução do Processo

```
<?php

function childs($id,$link) {
    $r = array();
    $categories_childs = "SELECT id from mdl_course_categories where parent = ".$id;
    if ($result = mysqli_query($link, $categories_childs)) {
        while($results = mysqli_fetch_array($result)) {
            $r[] = $results['id'];
            $r_childs = childs($results['id'],$link);
            if(count($r_childs) > 0) {
                foreach($r_childs as $childid) {
                    $r[] = $childid;
                }
            }
        }
    }
    return $r;
}

function courses($link,$categories) {
    $courses = array();
    foreach($categories as $cid) {
        $courses_query = "SELECT id from mdl_course where category = ".$cid;
        if($result = mysqli_query($link, $courses_query)) {
            while($course = mysqli_fetch_array($result)) {
                $courses[] = $course['id'];
            }
        }
    }
    return $courses;
}

?>
```

Figura 5.4: Introdução do Script de Backup

instância, recorrendo à documentação sobre *Moodle Repositories*[11], que permite associar uma pasta do sistema de ficheiros da máquina ao sistema de ficheiros da plataforma Moodle e para a qual deverão ser transferidos todas as cópias de segurança. De seguida, com um conjunto de comandos com recurso à tecnologia rsync² pode ser executada uma transferência dos ficheiros, como exemplificado na tabela 5.1. Concluída a transferência é então possível fazer o restauro dos backups pela interface de recuperação de cópias de segurança do Moodle.

²<http://linux.die.net/man/1/rsync>

<code>rsync -tavW --inplace username@moodle1.ul.pt:/pasta/onde/estao/backups/* pastaLocalTemporaria</code>
<code>rsync -tavW --inplace pastaLocalTemporaria/* username@moodle2.ul.pt:/pasta/preparada/com/repositories</code>

Tabela 5.1: Utilização exemplo rsync

```
$CFG->customfrontpageinclude=dirname(__FILE__) .'/local/degrees/frontpage.php';
```

Tabela 5.2: Configuração frontpage Moodle

5.2 Módulo Degrees

Este módulo tem como objectivo fazer alterações às regras de interface do Moodle. De acordo com a documentação oficial este tipo de alterações devem ser desenvolvidas segundo um *Local Plugin*³.

5.2.1 Concretização da Base de Dados

A base de dados do Módulo foi concretizada de acordo com a fig. 4.3 e com a documentação oficial para o desenvolvimento de base de dados de Módulos do Moodle[14]. Foi produzido um ficheiro no formato XML, de acordo com as regras da referida documentação, que definem as tabelas, respectivas entradas e tipos de valores.

5.2.2 Adaptação da interface

O Moodle a partir da sua versão 2.0 permite na sua configuração definir alterações na página principal⁴. Esta funcionalidade foi aproveitada para que seja possível a adaptação da nova interface. Assim sendo, no ficheiro *config.php* do Moodle foi inserida a linha como ilustra a tabela 5.2 que permite incluir na interface de utilização da página inicial a interface do módulo Degrees.

5.3 MOODLE-IS

Pretende-se concretizar alterações aos módulos *dif-elearning* e *MOODLE-IS* com o objectivo de tornar o SiGES capaz de projectar Unidades Curriculares em diferentes instâncias de uma plataforma e-learning, com principal foco no Moodle, aproveitando o processo de comunicação já existente. Para tal, foram feitas alterações na lógica do módulo *dif-elearning*, na utilização deste por parte do MOODLE-IS e nas interfaces que este último disponibiliza.

³https://docs.moodle.org/dev/Local_plugins

⁴<http://midact.com/content/how-create-custom-front-page-moodle-20>

5.3.1 Alterações dif-elearning

Mapa de Instâncias Como podemos observar pelo diagrama da fig. 4.7 a classe *ELearningConfiguration* tem um atributo *instanceMappings* onde é guardada toda a informação sobre o Mapa de Instâncias, isto é, quais os anos lectivos atribuídos a uma instância Moodle. Contudo, por exigência da arquitectura, este atributo tem que ser uma *String*. Para que o Mapa possa ser guardado neste atributo foi concretizado um método para a conversão do *HashMap*, que organiza uma lista de *Strings* representativas dos identificadores dos anos lectivos para cada *String* representativa do identificador da Instância, numa *String* em JSON e vice-versa com recurso a uma biblioteca já existente⁵.

5.3.2 Alterações MOODLE-IS

Processo de Projecção de Unidades Curriculares no Moodle Com o objectivo de concretizar as novas funcionalidades para os utilizadores do SiGES foram feitas algumas alterações no MOODLE-IS. Foi alterado o processo de projecção de Unidades Curriculares de forma a que seja possível verificar o cenário actual, isto é, se o MOODLE-IS está configurado para uma ou mais instâncias Moodle. Foram ajustadas as interfaces para configuração das projecções de Unidades Curriculares com o objectivo de permitir um utilizador escolher os anos lectivos referentes a uma determinada instância Moodle antes de iniciar o processo já existente. Foram também alteradas as interfaces de utilização que permitem executar este processo de modo a que seja possível escolher o ano lectivo que se pretende projectar.

5.4 Concretização da comunicação entre plataformas

5.4.1 Base de Dados de testes

Foi concretizada uma base de dados MySQL para os testes da comunicação do Moodle com o serviço web de acordo com o modelo da fig. 4.15 que permite simular a projecção de informação na plataforma de gestão académica. Para tal, foi utilizada uma ferramenta de nome *MySQL Workbench* que permite desenhar o modelo e, de forma rápida, exportar o seu código MySQL.

Recorrendo a essa ferramenta, o modelo de dados do serviço de testes foi exportado para o respectivo código MySQL e por sua vez foi importado num servidor de testes. Para acesso aos dados o servidor JBoss foi configurado com recurso a um JDBC (do inglês Java Database Connectivity)⁶ e foi desenvolvida um objecto Java, como representado na fig.

⁵<http://json-lib.sourceforge.net/apidocs/net/sf/json/package-summary.html>

⁶<http://www.oracle.com/technetwork/java/javase/jdbc/index.html>

5.5, com funções que permitem executar operações de escrita e leitura no referido modelo.

5.4.2 Concretização do Serviço Web

Partindo do desenho da comunicação e das suas mensagens concretizou-se um contracto WSDL de acordo com as normas definidas na W3C Note de 15 de Março de 2001⁷. Num primeiro passo foi desenhada uma definição dos WSDL Types para Classificações e Sumários.

Num segundo passo foi desenhado o contracto das WSDL Messages trocadas entre cliente e serviço por forma a conseguir um conjunto de mensagens para um serviço de Classificações e outro para um serviço de Sumários. Primeiro foram definidos os tipos XSD que definem uma Classificação, como ilustrado na fig. 5.6 e os tipos XSD que definem um Sumário como ilustrado na fig. 5.7. Para a comunicação foram definidos tipos XSD que representam as mensagens sobre Classificações, como ilustrado na fig. 5.8 e na fig. 5.9 e os tipos XSD que representam as mensagens sobre Sumários, como ilustrado na fig. 5.10 e na fig. 5.11.

Num último passo foram desenhados os pontos de entrada de cada serviço. Para cada um deles foram definidos os WSDL Ports, como ilustrados nas figuras 5.12 e 5.15, as respectivas WSDL Messages, como ilustradas nas figuras 5.13 e 5.16 e os WSDL Bindings, como ilustrados nas figuras 5.14 e 5.17.

Com a definição dos dois serviços, o contracto de WSDL foi utilizado para a criação do Serviço Web em Java num servidor JBoss com recurso a JAX-WS e de acordo com um processo semi-automático disponível pelo Eclipse publicado na documentação oficial do site jboss.org. Este processo disponibiliza um conjunto de objectos Java que concretizam o serviço web e permitem concretizar a implementação do negócio deste.⁸

De acordo com o contracto definido, duas classes de implementação foram criadas: SummaryEIIImpl.java e GradeEIIImpl.java. Ambas as classes disponibilizam métodos Java, de acordo com as operações identificadas, cujos argumentos são objectos que definem as mensagens de entrada e os valores de retorno são objectos que definem as respectivas mensagens de resposta. O corpo destes métodos é onde se concretizam as regras de negócio das mensagens a responder ao Moodle, no nosso caso as chamadas ao ServiceDAO.java que é responsável por operações de leitura e escrita na base de dados de teste.

⁷<http://www.w3.org/TR/wsdl>

⁸http://docs.jboss.org/tools/4.1.0.Final/en/ws_soap_reference/html/topdown.html

5.4.3 Módulo SummaryWS

Este módulo tem como objectivo acrescentar informação a uma Unidade Curricular do Moodle. De acordo com a documentação oficial este tipo de alterações devem ser desenvolvidas segundo um *Activity Module*⁹. Para a sua concretização utilizou-se uma estrutura já existente e disponível num repositório público¹⁰. Esta estrutura é um módulo exemplo do Moodle que organiza o código e os seus ficheiros para a concretização de Módulos deste tipo.

Concretização do cliente do serviço web Para a concretização do cliente foi utilizada a biblioteca para comunicação SOAP em PHP[16] que disponibiliza automaticamente funções de acordo com as operações definidas no contracto WSDL. Este cliente foi desenvolvido na classe SummaryWS desenhada de acordo com a figura 4.16.

5.4.4 Módulo GraderWS

Este módulo tem como objectivo exportar informação das Classificações duma Unidade Curricular do Moodle. De acordo com a documentação oficial este tipo de alterações devem ser desenvolvidas segundo um módulo *Gradebook Export*¹¹.

Concretização do cliente do Serviço Web Para a concretização do cliente foi utilizada a biblioteca para comunicação SOAP em PHP[16] que disponibiliza automaticamente funções de acordo com as operações definidas no contracto WSDL. Este cliente foi desenvolvido na classe GraderWS desenhada de acordo com a figura 4.16.

⁹https://docs.moodle.org/dev/Activity_modules

¹⁰https://github.com/moodlehq/moodle-mod_newmodule

¹¹https://docs.moodle.org/dev/Gradebook_export

ServiceDAO
<pre> -connect: Connection -statement: Statement -preparedStatement: PreparedStatement -resultSet: ResultSet -SERVICEWS_DATABASE: String = servicews -SERVICEWS_USER: String = root -SERVICEWS_PASS: String = password -SUMMARY_TABLE: String = Summary -SUMMARY_FIELD_ID: String = id -SUMMARY_FIELD_TEXT: String = summary -SUMMARY_FIELD_PRESENCES: String = presences -SUMMARY_FIELD_TEACHER: String = teacher_id -SUMMARY_FIELD_COURSE: String = course_id -CLASS_TABLE: String = course_class -CLASS_FIELD_ID: String = id -CLASS_FIELD_NOME: String = name -CLASS_FIELD_DATE: String = when -CLASS_FIELD_SUMMARY: String = summary_id -CLASS_FIELD_COURSE: String = course_id -STUDENTGRADE_TABLE: String = student_grade -STUDENTGRADE_FIELD_STUDENT: String = student_id -STUDENTGRADE_FIELD_COURSE: String = course_id -STUDENTGRADE_FIELD_COMPONENT: String = component_id -STUDENTGRADE_FIELD_GRADE: String = grade -COMPONENT_TABLE: String = component -COMPONENT_FIELD_ID: String = id -COMPONENT_FIELD_NAME: String = name -COMPONENT_FIELD_COURSE: String = course_id -COMPONENT_FIELD_MIN: String = min -COMPONENT_FIELD_MAX: String = max -STUDENT_TABLE: String = student -STUDENT_COURSE_TABLE: String = student_has_course -STUDENT_FIELD_ID: String = course_id -STUDENT_FIELD_COURSE: String = course_id -GRADE_TABLE: String = student_grade -GRADE_FIELD_STUDENT: String = student_id -GRADE_FIELD_COURSE: String = course_id -GRADE_FIELD_COMPONENT: String = component_id -GRADE_FIELD_GRADE: String = grade +ServiceDAO() +createSummary(sum:Summary): boolean +getClasses(course_id:String): List<Class> +getSummaryList(course_id:String): List<Summary> +createStudentGrade(course_id:String,grd:GradeMsg): boolean +getComponents(course_id:String): List<Component> +getStudents(course_id:String): List<String> +getGrades(course_id:String): List<GradeMsg> +getResultSet(statement:String): boolean +update(statement:String): boolean +parseDate(timedate:String): XMLGregorianCalendar +close(): void </pre>

Figura 5.5: ServiceDAO.java

```

1. <xsd:complexType name="gradeMsg">
2.     <xsd:sequence>
3.         <xsd:element name="component" type="msg:component"/>
4.         <xsd:element name="student_id" type="xsd:string"/>
5.         <xsd:element name="value" type="xsd:string"/>
6.     </xsd:sequence>
7. </xsd:complexType>
8.
9. <xsd:complexType name="scale">
10.     <xsd:sequence>
11.         <xsd:element name="min" type="xsd:string"/>
12.         <xsd:element name="max" type="xsd:string"/>
13.     </xsd:sequence>
14. </xsd:complexType>
15.
16. <xsd:complexType name="component">
17.     <xsd:sequence>
18.         <xsd:element name="id" type="xsd:string"/>
19.         <xsd:element name="name" type="xsd:string"/>
20.         <xsd:element name="scale" type="msg:scale"/>
21.     </xsd:sequence>
22. </xsd:complexType>

```

Figura 5.6: Definições de tipos XSD para Classificações

```

1. <xsd:complexType name="Class">
2.     <xsd:sequence>
3.         <xsd:element name="id" type="xsd:string" />
4.         <xsd:element name="value" type="xsd:string" />
5.         <xsd:element name="date" type="xsd:dateTime" />
6.     </xsd:sequence>
7. </xsd:complexType>
8.
9. <xsd:complexType name="Summary">
10.     <xsd:sequence>
11.         <xsd:element name="summary_id" type="xsd:string" />
12.         <xsd:element name="date" type="xsd:dateTime" />
13.         <xsd:element name="presences" type="xsd:int" />
14.         <xsd:element name="text" type="xsd:string" />
15.         <xsd:element name="course_id" type="xsd:string" />
16.         <xsd:element name="teacher_id" type="xsd:string" />
17.         <xsd:element name="class_list">
18.             <xsd:simpleType>
19.                 <xsd:list itemType="xsd:string" />
20.             </xsd:simpleType>
21.         </xsd:element>
22.     </xsd:sequence>
23. </xsd:complexType>

```

Figura 5.7: Definições de tipos XSD para Sumários

```

1. <xsd:complexType name="requestMsg">
2.   <xsd:sequence>
3.     <xsd:element name="course_id" type="xsd:string"/>
4.     <xsd:element name="session_id" type="xsd:string"/>
5.   </xsd:sequence>
6. </xsd:complexType>
7.
8. <xsd:complexType name="requestAnswerMsg">
9.   <xsd:sequence>
10.    <xsd:element name="code" type="xsd:int"/>
11.    <xsd:element name="answer" type="xsd:string"/>
12.    <xsd:element name="components" maxOccurs="unbounded" minOccurs="0">
13.      <xsd:complexType>
14.        <xsd:sequence>
15.          <xsd:element name="item" type="msg:component"/>
16.        </xsd:sequence>
17.      </xsd:complexType>
18.    </xsd:element>
19.    <xsd:element name="students">
20.      <xsd:simpleType>
21.        <xsd:list itemType="xsd:string"/>
22.      </xsd:simpleType>
23.    </xsd:element>
24.  </xsd:sequence>
25. </xsd:complexType>
26.
27. <xsd:complexType name="writeMsg">
28.   <xsd:sequence>
29.     <xsd:element name="course_id" type="xsd:string"/>
30.     <xsd:element name="session_id" type="xsd:string"/>
31.     <xsd:element name="grade" maxOccurs="unbounded" minOccurs="1">
32.       <xsd:complexType>
33.         <xsd:sequence>
34.           <xsd:element name="item" type="msg:gradeMsg"/>
35.         </xsd:sequence>
36.       </xsd:complexType>
37.     </xsd:element>
38.   </xsd:sequence>
39. </xsd:complexType>
40.
41. <xsd:complexType name="writeAnswerMsg">
42.   <xsd:sequence>
43.     <xsd:element name="code" type="xsd:int"/>
44.     <xsd:element name="answer" type="xsd:string"/>
45.   </xsd:sequence>
46. </xsd:complexType>

```

Figura 5.8: Definições de tipos XSD para Mensagens das Classificações

```

1. <xsd:complexType name="listAnswerMsg">
2.   <xsd:sequence>
3.     <xsd:element name="code" type="xsd:int"/>
4.     <xsd:element name="answer" type="xsd:string"/>
5.     <xsd:element name="grades" maxOccurs="unbounded" minOccurs="0">
6.       <xsd:complexType>
7.         <xsd:sequence>
8.           <xsd:element name="item" type="msg:gradeMsg"/>
9.         </xsd:sequence>
10.       </xsd:complexType>
11.     </xsd:element>
12.   </xsd:sequence>
13. </xsd:complexType>

```

Figura 5.9: Definições de tipos XSD para Mensagens das Classificações

```

1. <xsd:complexType name="requestMsg">
2.     <xsd:sequence>
3.         <xsd:element name="course_id" type="xsd:string" />
4.         <xsd:element name="session_id" type="xsd:string" />
5.     </xsd:sequence>
6. </xsd:complexType>
7.
8. <xsd:complexType name="requestAnswerMsg">
9.     <xsd:sequence>
10.        <xsd:element name="code" type="xsd:int" />
11.        <xsd:element name="answer" type="xsd:string" />
12.        <xsd:element name="classes" maxOccurs="unbounded"
13.            minOccurs="0">
14.            <xsd:complexType>
15.                <xsd:sequence>
16.                    <xsd:element type="msg:Class" name="item" />
17.                </xsd:sequence>
18.            </xsd:complexType>
19.        </xsd:element>
20.    </xsd:sequence>
21. </xsd:complexType>
22.
23. <xsd:complexType name="writeMsg">
24.     <xsd:sequence>
25.         <xsd:element name="session_id" type="xsd:string" />
26.         <xsd:element name="summaries" maxOccurs="unbounded"
27.             minOccurs="1">
28.             <xsd:complexType>
29.                 <xsd:sequence>
30.                     <xsd:element name="item" type="msg:Summary"/>
31.                 </xsd:sequence>
32.             </xsd:complexType>
33.         </xsd:element>
34.     </xsd:sequence>
35. </xsd:complexType>
36.
37. <xsd:complexType name="writeAnswerMsg">
38.     <xsd:sequence>
39.         <xsd:element name="code" type="xsd:int" />
40.         <xsd:element name="answer" type="xsd:string" />
41.     </xsd:sequence>
42. </xsd:complexType>

```

Figura 5.10: Definições de tipos XSD para Mensagens dos Sumários

```

1. <xsd:complexType name="listMsg">
2.     <xsd:sequence>
3.         <xsd:element name="session_id" type="xsd:string" />
4.         <xsd:element name="course_id" type="xsd:string" />
5.     </xsd:sequence>
6. </xsd:complexType>
7.
8. <xsd:complexType name="listAnswerMsg">
9.     <xsd:sequence>
10.        <xsd:element name="code" type="xsd:int" />
11.        <xsd:element name="answer" type="xsd:string" />
12.        <xsd:element name="list" maxOccurs="unbounded" minOccurs="0">
13.            <xsd:complexType>
14.                <xsd:sequence>
15.                    <xsd:element name="item" type="msg:Summary" />
16.                </xsd:sequence>
17.            </xsd:complexType>
18.        </xsd:element>
19.        <xsd:element name="classes" maxOccurs="unbounded"
20.            minOccurs="1">
21.            <xsd:complexType>
22.                <xsd:sequence>
23.                    <xsd:element name="item" type="msg:Class" />
24.                </xsd:sequence>
25.            </xsd:complexType>
26.        </xsd:element>
27.    </xsd:sequence>
28. </xsd:complexType>

```

Figura 5.11: Definições de tipos XSD para Mensagens dos Sumários

```

1  <?xml version="1.0"?>
2  <wsdl:definitions name="GraderWS"
3      xmlns:wsdl="http://schemas.xmlsoap.org/wsdl/" xmlns:soap="http://schemas.xmlsoap.org/wsdl/soap/"
4      xmlns:schema="http://localhost:8080/services/GradeEI?wsdl"
5      xmlns:tns="http://localhost:8080/services/GradeEI"
6      targetNamespace="http://localhost:8080/services/GradeEI">
7
8      <wsdl:documentation>
9          Description of SIS Grade Web Service
10     </wsdl:documentation>
11
12     <wsdl:portType name="GradeEI">
13         <wsdl:documentation>Grade operations</wsdl:documentation>
14
15         <wsdl:operation name="gradeRequest">
16             <wsdl:documentation>Request info to write Grade</wsdl:documentation>
17             <wsdl:input message="tns:request" />
18             <wsdl:output message="tns:requestAnswer" />
19         </wsdl:operation>
20
21         <wsdl:operation name="gradeWrite">
22             <wsdl:documentation>Write a Grade</wsdl:documentation>
23             <wsdl:input message="tns:write" />
24             <wsdl:output message="tns:writeAnswer" />
25         </wsdl:operation>
26
27         <wsdl:operation name="gradeList">
28             <wsdl:documentation>Get a grade list</wsdl:documentation>
29             <wsdl:input message="tns:list" />
30             <wsdl:output message="tns:listAnswer" />
31         </wsdl:operation>
32     </wsdl:portType>
33

```

Figura 5.12: Definições de WSDL Port Types para Classificações

```

1  <wsdl:message name="request">
2      <wsdl:part element="schema:request" name="parameters" />
3  </wsdl:message>
4
5  <wsdl:message name="requestAnswer">
6      <wsdl:part element="schema:requestAnswer" name="answer" />
7  </wsdl:message>
8
9  <wsdl:message name="write">
10     <wsdl:part element="schema:write" name="parameters" />
11 </wsdl:message>
12
13 <wsdl:message name="writeAnswer">
14     <wsdl:part element="schema:writeAnswer" name="answer" />
15 </wsdl:message>
16
17 <wsdl:message name="list">
18     <wsdl:part element="schema:list" name="parameters" />
19 </wsdl:message>
20
21 <wsdl:message name="listAnswer">
22     <wsdl:part element="schema:listAnswer" name="answer" />
23 </wsdl:message>

```

Figura 5.13: Definições de WSDL Messages para Classificações


```

1      <wsdl:types>
2          <xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema">
3              <xsd:import namespace="http://localhost:8080/servicews/GradeEI?wsdl"
4                  schemaLocation="gradeSchema.xsd" />
5          </xsd:schema>
6      </wsdl:types>
7
8      <wsdl:binding name="GradeBinding" type="tns:GradeEI">
9          <soap:binding style="document"
10              transport="http://schemas.xmlsoap.org/soap/http" />
11
12          <wsdl:operation name="gradeRequest">
13              <soap:operation
14                  soapAction="http://localhost:8080/servicews/GradeEI/gradeRequest" />
15              <wsdl:input>
16                  <soap:body use="literal" />
17              </wsdl:input>
18              <wsdl:output>
19                  <soap:body use="literal" />
20              </wsdl:output>
21          </wsdl:operation>
22
23          <wsdl:operation name="gradeWrite">
24              <soap:operation soapAction="http://localhost:8080/servicews/GradeEI/gradeWrite" />
25              <wsdl:input>
26                  <soap:body use="literal" />
27              </wsdl:input>
28              <wsdl:output>
29                  <soap:body use="literal" />
30              </wsdl:output>
31          </wsdl:operation>
32
33          <wsdl:operation name="gradeList">
34              <soap:operation soapAction="http://localhost:8080/servicews/GradeEI/gradeList" />
35              <wsdl:input>
36                  <soap:body use="literal" />
37              </wsdl:input>
38              <wsdl:output>
39                  <soap:body use="literal" />
40              </wsdl:output>
41          </wsdl:operation>
42      </wsdl:binding>
43
44      <wsdl:service name="GradeEI">
45          <wsdl:port binding="tns:GradeBinding" name="GradeEndpoint">
46              <soap:address location="http://localhost:8080/servicews/GradeEI" />
47          </wsdl:port>
48      </wsdl:service>
49
50  </wsdl:definitions>

```

Figura 5.14: Definições de WSDL Bindings para Classificações

```

1  <?xml version="1.0"?>
2  <wsdl:definitions name="SummaryWS"
3      xmlns:wsdl="http://schemas.xmlsoap.org/wsdl/" xmlns:soap="http://schemas.xmlsoap.org/wsdl/soap/"
4      xmlns:schema="http://localhost:8080/servicews/SummaryEI?wsdl"
5      xmlns:tns="http://localhost:8080/servicews/SummaryEI"
6      targetNamespace="http://localhost:8080/servicews/SummaryEI">
7
8      <wsdl:documentation>
9          Description of SIS Summary Web Service
10     </wsdl:documentation>
11
12     <wsdl:portType name="SummaryEI">
13         <wsdl:documentation>Summary operations</wsdl:documentation>
14
15         <wsdl:operation name="summaryRequest">
16             <wsdl:documentation>Request info to write summary
17             </wsdl:documentation>
18             <wsdl:input message="tns:request" />
19             <wsdl:output message="tns:requestAnswer" />
20         </wsdl:operation>
21
22         <wsdl:operation name="summaryWrite">
23             <wsdl:documentation>Write summary</wsdl:documentation>
24             <wsdl:input message="tns:write" />
25             <wsdl:output message="tns:writeAnswer" />
26         </wsdl:operation>
27
28         <wsdl:operation name="summaryList">
29             <wsdl:documentation>Get a summary list</wsdl:documentation>
30             <wsdl:input message="tns:list" />
31             <wsdl:output message="tns:listAnswer" />
32         </wsdl:operation>
33     </wsdl:portType>
34

```

Figura 5.15: Definições de WSDL Port Types para Sumários

```

1  <wsdl:message name="request">
2      <wsdl:part element="schema:request" name="parameters" />
3  </wsdl:message>
4
5  <wsdl:message name="requestAnswer">
6      <wsdl:part element="schema:requestAnswer" name="parameters" />
7  </wsdl:message>
8
9  <wsdl:message name="write">
10     <wsdl:part element="schema:write" name="parameters" />
11 </wsdl:message>
12
13 <wsdl:message name="writeAnswer">
14     <wsdl:part element="schema:writeAnswer" name="parameters" />
15 </wsdl:message>
16
17 <wsdl:message name="list">
18     <wsdl:part element="schema:list" name="parameters" />
19 </wsdl:message>
20
21 <wsdl:message name="listAnswer">
22     <wsdl:part element="schema:listAnswer" name="parameters" />
23 </wsdl:message>

```

Figura 5.16: Definições de WSDL Messages para Sumários

```

1      <wsdl:types>
2          <xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema">
3              <xsd:import namespace="http://localhost:8080/servicews/SummaryEI?wsdl"
4                  schemaLocation="summarySchema.xsd" />
5          </xsd:schema>
6      </wsdl:types>
7
8      <wsdl:binding name="SummaryBinding" type="tns:SummaryEI">
9          <soap:binding style="document"
10              transport="http://schemas.xmlsoap.org/soap/http" />
11
12          <wsdl:operation name="summaryRequest">
13              <soap:operation
14                  soapAction="http://localhost:8080/servicews/SummaryEI/summaryRequest" />
15              <wsdl:input>
16                  <soap:body use="literal" />
17              </wsdl:input>
18              <wsdl:output>
19                  <soap:body use="literal" />
20              </wsdl:output>
21          </wsdl:operation>
22
23          <wsdl:operation name="summaryWrite">
24              <soap:operation
25                  soapAction="http://localhost:8080/servicews/SummaryEI/summaryWrite" />
26              <wsdl:input>
27                  <soap:body use="literal" />
28              </wsdl:input>
29              <wsdl:output>
30                  <soap:body use="literal" />
31              </wsdl:output>
32          </wsdl:operation>
33
34          <wsdl:operation name="summaryList">
35              <soap:operation
36                  soapAction="http://localhost:8080/servicews/SummaryEI/summaryList" />
37              <wsdl:input>
38                  <soap:body use="literal" />
39              </wsdl:input>
40              <wsdl:output>
41                  <soap:body use="literal" />
42              </wsdl:output>
43          </wsdl:operation>
44
45      </wsdl:binding>
46
47      <wsdl:service name="SummaryEI">
48          <wsdl:port binding="tns:SummaryBinding" name="SummaryEndpoint">
49              <soap:address location="http://localhost:8080/servicews/SummaryEI" />
50          </wsdl:port>
51      </wsdl:service>
52
53  </wsdl:definitions>

```

Figura 5.17: Definições de WSDL Bindings para Sumários

Capítulo 6

Avaliação

6.1 Histórico Moodle

Transferência de cópias de segurança Para a validação do processo de transferência de cópias de segurança da instância em produção para o histórico Moodle foram feitas algumas execuções dos scripts concretizados e foram feitas comparações dos conteúdos das unidades curriculares nas duas instâncias. Identificaram-se cenários de erro como o caso de unidades curriculares que utilizam módulos Moodle na instância em produção que não estão instalados na instância do histórico.

Assim se concluiu que é necessário um processo de validação da instância do histórico onde se verifica que todos os módulos presentes no Moodle em produção também estão presentes no Moodle histórico. Deu-se o processo da concretização do histórico Moodle por concluído uma vez validada a comparação dos conteúdos em ambas as instâncias.

6.2 Apresentação e Organização das Unidades Curriculares

Testes ao módulo Degrees Foram feitos testes com a utilização das interfaces de gestão por fim a avaliar as funcionalidades do módulo. Verificaram-se as funcionalidades de manipulação e de apresentação dos dados. Foi avaliado se uma unidade curricular ficaria acessível por diferentes categorias, se seria apresentável com nomes diferentes e se a ordem dos conteúdos era a expectável pelo utilizador.

Deu-se por concluída a concretização deste módulo com a validação destes testes que permitiram concluir que será possível cumprir os novos requisitos de apresentação e organização das unidades curriculares no Moodle.

6.3 MOODLE-IS com Projecção Multi-Instâncias

Configuração de instâncias Moodle por ano lectivo Utilizaram-se as novas interfaces do MOODLE-IS para a criação e configuração de várias instâncias Moodle para diferentes anos lectivos. Experimentaram-se vários cenários de teste em que um Moodle tem apenas um ano lectivo, um Moodle tem mais do que um ano lectivo, duas instâncias de Moodle têm diferentes conjuntos de anos lectivos, um Moodle deixaria de ter um dos seus anos lectivos e, por fim, um ano lectivo transitaria de um Moodle para o outro.

As alterações deram-se por concluídas quando os testes foram validados no correcto funcionamento da interface com os resultados expectáveis.

6.4 Comunicação entre plataformas

Testes ao Serviço Web Para a concretização dos testes ao serviço foi utilizada a biblioteca para comunicação SOAP em PHP[16] que disponibiliza automaticamente funções de acordo com as operações definidas no contracto WSDL. Foi preparado um conjunto de chamadas às funcionalidades disponiveis no contracto do serviço e foram avaliados os parâmetros de resposta para diferentes contextos, ou seja, foram variados os dados presentes na base de dados do serviço.

Verificou-se que, em diferentes contextos, as estruturas de dados em PHP das respostas variavam, isto é, ao pedir uma lista de sumários a resposta de conter apenas um sumário era diferente do que quando continha mais do que um ou nenhum sumário. Estes testes permitiram validar o funcionamento da comunicação com o contracto disposto e concluir que seria necessário uma adaptação do cliente do serviço web na interpretação das respostas para os vários cenários mencionados.

Módulo SummaryWS Foram feitos testes a esta concretização com recurso às interfaces de utilização por forma a avaliar a gestão e apresentação dos conteúdos do serviço web em vários cenários. Foram avaliados os casos em que não existem sumários a apresentar, existe apenas um sumário ou existe mais do que um sumário e foi validada a criação e apresentação de um sumário numa ou mais turmas. Com estes testes foi possível validar a adaptação da interface aos vários cenários mencionados. A concretização do módulo SummaryWS deu-se por concluída após a validação dos dados apresentados na interface.

Módulo GraderWS Foram feitos testes a esta concretização com recurso às interfaces de utilização por forma a avaliar a exportação de classificações em vários cenários. Foram projectadas algumas componentes de avaliação de forma correcta e incorrecta no que diz respeito a validação da escala, foram exportadas classificações de vários alunos e foram alteradas projecções de classificações por forma a alterar os dados no serviço de testes.

Com estes testes foi possível validar a adaptação da interface aos vários cenários mencionados. A concretização do módulo GraderWS deu-se por concluída após a validação dos dados apresentados.

Capítulo 7

Conclusão

Face aos objectivos apresentados concluiu-se que é possível a concretização de soluções para os requisitos levantados com recurso às tecnologias propostas. Os testes feitos serviram para validar o processo de concretização e de implementação, concluindo que estes são exequíveis e que cumprem os objectivos de redução do fluxo de trabalho dos utilizadores e a integração de informação entre as plataformas de apoio à aprendizagem e de gestão académica da Universidade de Lisboa.

7.1 Histórico Moodle

Foi conseguida a criação de um Histórico de Unidades Curriculares de anos lectivos anteriores e a implementação de uma estratégia facilitadora na sua manutenção.

Uma vez que se pretende implementar uma nova plataforma Moodle na versão 2.4 na Universidade de Lisboa o histórico deve acompanhar a mesma versão e, por isso, deve ser feito um upgrade da instância de histórico. No que diz respeito à migração dos conteúdos o processo pode ainda ser melhorado com o auxílio do novo mecanismo de projecção de Unidades Curriculares do MOODLE-IS.

É possível desenhar um mecanismo que, através da informação referente às projecções das Unidades Curriculares em várias instâncias, execute um processo automático de transferência de conteúdos entre plataformas Moodle.

7.2 Comunicação entre Moodle e SiGES

Foi conseguida a concretização de uma solução que resolve a inconveniente separação destas duas plataformas com recurso a tecnologias e ferramentas existentes. Dado que foi concretizado o serviço web para as funcionalidades apresentadas, é possível implementá-lo nas plataformas SiGES em produção por fim a concretizar as projecções indicadas.

Por forma a que esta comunicação seja implementada, é também necessária a concretização de técnicas de segurança como sejam a utilização de comunicação segura por via de

HTTPS e processos de autenticação dos serviços ou utilizadores.

Conseguiu-se também que esta solução possa ser aplicável a futuros requisitos levantados e a outras plataformas do mesmo género. Este trabalho deu-se por concluído na sua fase de concretização e validação num cenário de testes e apresenta resultados muito próximos da sua fase de implementação.

Bibliografia

- [1] Digitalis.pt. *DIF - Digitalis Framework*. Abr. de 2012. URL: <http://development.digitalis.pt/apache2-default/dif2/index.html>.
- [2] Digitalis.pt. *DIF Core - Digitalis Framework Core Module*. Abr. de 2012. URL: <http://development.digitalis.pt/apache2-default/dif2/dif-core/index.html>.
- [3] Digitalis.pt. *DIF Model - Digitalis Framework Model Module*. Abr. de 2012. URL: <http://development.digitalis.pt/apache2-default/dif2/dif-model/index.html>.
- [4] Digitalis.pt. *DIF Modules - Digitalis Framework Model Additional Modules*. Abr. de 2012. URL: <http://development.digitalis.pt/apache2-default/dif2/dif-modules/index.html>.
- [5] Digitalis.pt. *DIF Presentation - Digitalis Framework Presentation Module*. Abr. de 2012. URL: <http://development.digitalis.pt/apache2-default/dif2/dif-presentation/index.html>.
- [6] Digitalis.pt. *Digi-Utils - Digitalis Utility Pack*. Abr. de 2012. URL: <http://development.digitalis.pt/apache2-default/digi-utils/index.html>.
- [7] Moogles.org. *Moodle Cron*. Abr. de 2014. URL: <https://docs.moodle.org/24/en/Cron>.
- [8] Moogles.org. *Moodle Developer Documentation*. Ago. de 2014. URL: https://docs.moodle.org/dev/Developer_documentation.
- [9] Moogles.org. *Moodle Grade export*. Ago. de 2014. URL: http://docs.moodle.org/23/en/Grade_export.
- [10] Moogles.org. *Moodle Plugins*. Jul. de 2014. URL: <https://docs.moodle.org/dev/Plugins>.
- [11] Moogles.org. *Moodle Repositories*. Fev. de 2012. URL: <https://docs.moodle.org/22/en/Repositories>.
- [12] Moogles.org. *Moodle User Interface Guidelines*. Ago. de 2011. URL: https://docs.moodle.org/dev/Moodle_User_Interface_Guidelines.
- [13] Moogles.org. *Moodle Web Services*. Abr. de 2012. URL: http://docs.moodle.org/dev/Web_services.
- [14] Moogles.org. *Moodle XMLDB defining an XML structure*. Mai. de 2011. URL: https://docs.moodle.org/dev/XMLDB_defining_an_XML_structure.

- [15] PHP.net. *argv*. Jan. de 2014. URL: <http://php.net/manual/en/reserved.variables.argv.php>.
- [16] PHP.net. *The SoapClient class*. Jan. de 2014. URL: <http://php.net/manual/en/class.soapclient.php>.
- [17] Samba.org. *rsync*. Jun. de 2014. URL: <http://rsync.samba.org/ftp/rsync/rsync.html>.